



TUGAS AKHIR - KI141502

KLASIFIKASI EEG ALKOHOLIK MENGGUNAKAN WAVELET PACKET DECOMPOSITION, PRINCIPAL COMPONENT ANALYSIS, DAN KOMBINASI GENETIC ALGORITHM DAN NEURAL NETWORK

MUHAMMAD SADDAM
NRP 5113100187

Dosen Pembimbing I
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

**KLASIFIKASI EEG ALKOHOLIK MENGGUNAKAN
WAVELET PACKET DECOMPOSITION,
PRINCIPAL COMPONENT ANALYSIS, DAN
KOMBINASI GENETIC ALGORITHM DAN
NEURAL NETWORK**

**MUHAMMAD SADDAM
NRP 5113100187**

**Dosen Pembimbing I
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D**

**Dosen Pembimbing II
Dini Adni Navastara, S.Kom. M.Sc.**

**Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESES - KI141502

**CLASSIFICATION OF ALCOHOLIC EEG USING
WAVELET PACKET DECOMPOSITION,
PRINCIPAL COMPONENT ANALYSIS, AND
COMBINATION OF GENETIC ALGORITHM AND
NEURAL NETWORK**

**MUHAMMAD SADDAM
NRP 5113100187**

First Advisor

Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D

Second Advisor

Dini Adni Navastara, S.Kom., M.Sc.

**Department of Informatics
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017**

(Halaman ini sengaja dikosongkan)



LEMBAR PENGESAHAN

KLASIFIKASI EEG ALKOHOLIK MENGGUNAKAN WAVELET PACKET DECOMPOSITION, PRINCIPAL COMPONENT ANALYSIS, DAN KOMBINASI GENETIC ALGORITHM DAN NEURAL NETWORK

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

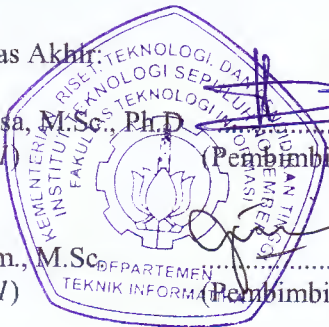
Oleh:

MUHAMMAD SADDAM

NRP: 5113100187

Disetujui oleh Pembimbing Tugas Akhir:

1. Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D
(NIP. 194908231976032001) (Pembimbing 1)
2. Dini Adni Navastara, S.Kom., M.Sc
(NIP. 198510172015042001) (Pembimbing 2)



**SURABAYA
MARET, 2017**

(Halaman ini sengaja dikosongkan)

KLASIFIKASI EEG ALKOHOLIK MENGGUNAKAN WAVELET PACKET DECOMPOSITION, PRINCIPAL COMPONENT ANALYSIS, DAN KOMBINASI GENETIC ALGORITHM DAN NEURAL NETWORK

Nama Mahasiswa : MUHAMMAD SADDAM
NRP : 5113100187
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Prof. Ir. Handayani Tjandrasa,
M.Sc., Ph.D
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

Abstrak

Alkoholisme adalah gangguan yang ditandai oleh konsumsi berlebihan dan ketergantungan pada alkohol. Terdapat bermacam cara untuk mendeteksi apakah seorang pasien telah kecanduan alkohol, salah satunya dengan deteksi otak menggunakan Electroencephalographic (EEG).

Sinyal EEG secara luas dan klinis digunakan untuk melakukan deteksi gangguan otak pada dunia kesehatan. Akan tetapi, sinyal yang dihasilkan oleh EEG perlu dipersiapkan untuk dilakukan proses agar dapat mendeteksi kelainan otak secara otomatis. Oleh karena itu, perlu adanya metode praproses untuk ekstraksi fitur yang tepat agar mendapatkan karakteristik yang tersimpan secara implisit dari sinyal EEG tersebut.

Tugas Akhir ini akan mengimplementasikan metode Wavelet Packet Decomposition (WPD) untuk ekstraksi fitur, Principal Component Analysis (PCA) untuk reduksi dimensi, dan Neural Network yang dioptimasi dengan metode Genetic Algorithm dalam pencarian bobot dan bias optimal untuk klasifikasi data alkoholik dan normal. Data uji coba yang digunakan dibagi menjadi 2 yaitu, dataset 1 terdiri dari 60 training dan 40 testing serta dataset 2 terdiri dari 120 training dan 40 testing. Berdasarkan hasil uji coba, rata-rata akurasi terbaik

didapatkan dari dataset 1 sebesar 94.00% dengan dekomposisi 3 level, penggunaan fitur 30%, dan klasifikasi menggunakan Kombinasi Neural Network dan Genetic Algorithm dengan learning rate 0.1 dan nilai alpha pada proses crossover 0.9.

Kata kunci: Alcoholism, EEG, Wavelet Packet Decomposition, Principal Component Analysis, Neural Network, Genetic Algorithm

CLASSIFICATION OF ALCOHOLIC EEG USING WAVELET PACKET DECOMPOSITION, PRINCIPAL COMPONENT ANALYSIS, AND COMBINATION OF GENETIC ALGORITHM AND NEURAL NETWORK

Student's Name : MUHAMMAD SADDAM
Student's ID : 5113100187
Department : Teknik Informatika FTIF-ITS
First Advisor : Prof. Ir. Handayani Tjandrasa, M.Sc.,
Ph.D
Second Advisor : Dini Adni Navastara, S.Kom., M.Sc.

Abstract

Alcoholism is a disorder characterized by excessive consumption and dependence on alcohol. There are various ways to detect whether a patient is addicted to alcohol, one of them by brain detection using electroencephalographic (EEG).

EEG signals are widely and clinically used for detection of brain disorders in the medical world. However, the signals generated by the EEG should be prepared to do further processing to detect brain abnormalities automatically. Initially, preprocessing method is needed for proper feature extraction, in order to obtain characteristics that are stored implicitly in the EEG signal.

This undergraduate thesis implements Wavelet Packet Decomposition (WPD) method for feature extraction, Principal Component Analysis (PCA) for dimension reduction, and Back Propagation Neural Network optimized with Genetic Algorithm to get optimal weights and biases for alcohol addiction classification. The EEG data used are divided into two: the first dataset consists of 60 training data and 40 testing data, and the second dataset consists of 120 training data and 40 testing data. Based on the experiment results, the first dataset gives best performance with 94.00% accuracy with decomposition of 3 levels, taking 30% of the

features, and classification using Neural Network and Genetic Algorithm with learning rate of 0.1 and crossover alpha value of 0.9.

Keywords : Alcoholism, EEG, Wavelet Packet Decomposition, Principal Component Analysis, Neural Network, Genetic Algorithm

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“Klasifikasi EEG Alkoholik Menggunakan Wavelet Packet Decomposition, Principal Component Analysis, dan Kombinasi Genetic Algorithm dan Neural Network”

Terselesaikannya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan banyak pihak, Oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan kepada:

1. Allah SWT serta junjungan Nabi Muhammad SAW, karena limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Teknik Informatika ITS.
2. Ayah dan Ibu penulis, Dahlan dan Hasniaty yang tiada hentinya memberikan dukungan doa, moral, dan material kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Kakak penulis, Muhammad Fahlevi dan Muhammad Yasser serta keluarga tercinta yang telah memberikan dukungan dan semangatnya kepada penulis.
4. Ibu Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D. dan Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku pembimbing I dan II yang telah membimbing dan memberikan motivasi, nasihat, dan bimbingan dalam menyelesaikan Tugas Akhir ini.
5. Bapak Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku kepala jurusan Teknik Informatika ITS dan segenap dosen

dan karyawan Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa studi di Teknik Informatika ITS.

6. Sahabat penulis yang tidak dapat disebutkan satu per satu yang selalu membantu, menghibur, menjadi tempat bertukar ilmu serta pembelajaran baru dan berjuang bersama-sama penulis.
7. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis kedepannya. Selain itu, penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum.

Surabaya, Maret 2017

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT	ix
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Permasalahan	3
1.4 Tujuan	3
1.5 Manfaat.....	4
1.6 Metodologi	4
1.6.1 Penyusunan Proposal Tugas Akhir.....	4
1.6.2 Studi Literatur.....	4
1.6.3 Implementasi Perangkat Lunak	5
1.6.4 Pengujian dan Evaluasi	5
1.6.5 Penyusunan Buku.....	5
1.7 Sistematika Penulisan Laporan	6
BAB II TINJAUAN PUSTAKA	9
2.1 Alkoholisme	9
2.2 Electroencephalography (EEG).....	10
2.3 Wavelet Packet Decomposition (WPD)	12
2.4 Principal Component Analysis (PCA)	14
2.5 Kombinasi Genetic Algorithm dan Neural Network.....	15
2.5.1 Back Propagation Neural Network.....	16
2.5.2 Genetic Algorithm (GA)	17
2.6 Confusion Matrix	22
BAB III PERANCANGAN SISTEM.....	25
3.1 Data	25
3.1.1 Data Masukan.....	25

3.1.2	Data Keluaran.....	27
3.2	Desain Umum Sistem	27
3.2.1	Wavelet Packet Decomposition (WPD)	29
3.2.2	Principal Component Analysis (PCA).....	31
3.3	Kombinasi Genetic Algorithm dan Neural Network.....	32
3.3.1	Pembangunan dan Pengaturan Network.....	33
3.3.2	Pengaturan Parameter Awal Genetic Algorithm	35
BAB IV	IMPLEMENTASI.....	37
4.1	Lingkungan Implementasi	37
4.2	Implementasi	37
4.2.1	Implementasi Pemrosesan Data Sebelum Praprocessing.....	38
4.2.2	Implementasi Wavelet Packet Decomposition (WPD)	39
4.2.3	Implementasi Principal Component Analysis (PCA)	41
4.2.4	Implementasi Kombinasi Genetic Algorithm dan Neural Network	42
BAB V	UJI COBA DAN EVALUASI.....	49
5.1	Lingkungan Pengujian.....	49
5.2	Data Pengujian	50
5.3	Skenario Uji Coba	50
5.3.1	Skenario Uji Coba Perhitungan Performa Berdasarkan Fitur yang dihasilkan PCA	51
5.3.2	Skenario Uji Coba Performa Metode Genetic Algorithm Pada Neural Network.....	52
5.3.3	Skenario Uji Coba Perhitungan Performa Berdasarkan Learning Rate NN.....	55
5.3.4	Skenario Uji Coba Perhitungan Performa kombinasi GA-NN Berdasarkan Nilai alpha.....	56
5.3.5	Skenario Uji Coba Perbandingan Performa dengan Penelitian Sebelumnya	57
5.4	Evaluasi Umum Skenario Uji Coba	58
BAB VI	KESIMPULAN DAN SARAN	61
6.1	Kesimpulan.....	61

6.2	Saran.....	62
DAFTAR PUSTAKA		63
LAMPIRAN.....		67
BIODATA PENULIS.....		75

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Perbedaan Otak Alkoholik dan Normal.....	10
Gambar 2.2. Konfigurasi EEG	11
Gambar 2.3 Contoh Rekaman EEG	12
Gambar 2.4 Topologi WPD.....	13
Gambar 2.5 Topologi PCA.....	15
Gambar 2.6 Ilustrasi Back Propagation Neural Network	17
Gambar 2.7 Ilustrasi Arithmetic Crossover.....	19
Gambar 2.8 Ilustrasi Mutasi	20
Gambar 2.9 Diagram alir proses GA.....	21
Gambar 3.1 Contoh Data Awalan	26
Gambar 3.2 Diagram Alir Keseluruhan Sistem.....	28
Gambar 3.3 Langkah Dekomposisi Sinyal WPD	30
Gambar 3.4 Langkah Reduksi Dimensi PCA.....	32
Gambar 3.5 Diagram Alir Klasifikasi GA-NN.....	34
Gambar 3.6 Representasi Kromosom.....	35
Gambar 3.7 Ilustrasi Pembuatan Network.....	36
Gambar 5.1 Perbandingan Akurasi BPNN dan GA-NN	53
Gambar 5.2 Grafik MSE BPNN.....	54
Gambar 5.3 Grafik MSE Kombinasi GA-NN	54

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2.1. Jenis Sinyal EEG	11
Tabel 2.2 Confusion Matrix dua kelas	22
Tabel 4.1 Spesifikasi lingkungan implementasi	37
Tabel 5.1 Spesifikasi Lingkungan Uji Coba.....	49
Tabel 5.2 Jumlah Fitur yang Dihasilkan PCA.....	51
Tabel 5.3 Perhitungan Performa PCA	52
Tabel 5.4 Perbandingan Performa BPNN dan Kombinasi	53
Tabel 5.5 Perhitungan Performa GA-NN Berdasarkan Learning Rate.....	56
Tabel 5.6 Perhitungan performa GA-NN Berdasarkan Nilai alpha (α)	57

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4.1 Pemrosesan Data Awal.....	38
Kode Sumber 4.2 Kode program penghapusan sinyal	39
Kode Sumber 4.3 Dekomposisi sinyal WPD	40
Kode Sumber 4.4 Dekomposisi sinyal WPD Lanjutan	41
Kode Sumber 4.5 Reduksi dimensi PCA	42
Kode Sumber 4.6 Pembangunan Neural Network	43
Kode Sumber 4.7 Selection	44
Kode Sumber 4.8 Crossover.....	45
Kode Sumber 4.9 Mutasi.....	45
Kode Sumber 4.10 Evaluasi GA	46
Kode Sumber 4.11 Evaluasi Data Testing.....	47

(Halaman sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Alkoholisme adalah gangguan yang ditandai oleh konsumsi berlebihan dan ketergantungan pada alkohol. Alkohol sendiri sudah menjadi masalah kesehatan yang umum dan menyerang 10% dari populasi dunia [1]. Kecanduan alkohol dapat menyebabkan beberapa faktor negatif dan resiko yang mencakup gangguan psikologis, seperti mengakibatkan stres, depresi, gangguan kecemasan, gangguan bipolar, atau skizofrenia. Terdapat bermacam cara untuk mendeteksi apakah seorang pasien telah kecanduan alkohol, salah satunya dengan deteksi otak menggunakan *Electroencephalographic* (EEG) [2].

Sinyal EEG secara luas dan klinis digunakan untuk melakukan deteksi gangguan otak pada dunia kesehatan. Beberapa studi mengatakan bahwa pencatatan aktivitas dari aktivitas listrik otak melalui pencatatan EEG merupakan salah satu teknik yang penting untuk diagnosis penyakit saraf [3]. Akan tetapi, sinyal yang dihasilkan oleh EEG dianggap belum siap untuk dilakukan proses lebih lanjut agar dapat mendeteksi kelainan otak. Oleh karena itu, perlu adanya metode praproses untuk ekstraksi fitur yang tepat agar mendapatkan karakteristik yang tersimpan secara implisit dari sinyal EEG tersebut. Terdapat berbagai macam metode praproses data EEG, salah satunya adalah *Wavelet Transform* yang mendekomposisi sinyal menjadi berbagai band frekuensi sehingga menghasilkan fitur baru dan *Principal Component Analysis* (PCA) untuk mereduksi dimensi dari vektor fitur yang didapatkan.

Telah banyak metode klasifikasi yang sudah digagaskan sebelumnya untuk dapat mendeteksi penyakit berdasarkan sinyal EEG, salah satunya adalah *Back Propagation Neural Network* (BPNN) [4]. BPNN merupakan algoritma pelatihan terbimbing (*supervised learning*) dari domain *Neural Network* yang

menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur. BPNN memiliki beberapa kekurangan, salah satunya memiliki sejumlah parameter yang cukup banyak dan harus dimasukkan secara manual. Dominasi jumlah parameter pada *Neural Network* terletak pada nilai bobot dan bias. Nilai bobot dan bias awal yang secara *default* diatur acak inilah yang mempunyai pengaruh besar terhadap kinerja *Neural Network*. Sehingga dibutuhkan optimasi pada bobot dan bias awal untuk menunjang kinerja *Neural Network* secara optimal. Salah satu metode yang dapat mengoptimasi kinerja BPNN adalah dengan mengimplementasikan algoritma evolusi, *Genetic Algorithm* (GA) pada metode BPNN.

Oleh karena itu, pada Tugas Akhir ini akan dibangun sebuah perangkat lunak untuk klasifikasi EEG alkoholik menggunakan metode *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan *Back Propagation Neural Network* yang dioptimasi menggunakan metode *Genetic Algorithm* (GA). Sebelumnya, data dari EEG akan melewati 3 tahap utama yaitu, praproses menggunakan metode *Wavelet Packet Decomposition* (WPD) untuk dekomposisi dan ekstraksi fitur sinyal, *Principal Component Analysis* (PCA) untuk proses reduksi dimensi fitur dari data EEG, dan selanjutnya akan dilakukan proses klasifikasi menggunakan metode kombinasi *Genetic Algorithm* dan *Neural Network*. Diharapkan dengan metode yang diajukan pada Tugas Akhir ini dapat menghasilkan nilai akurasi yang lebih baik dari penelitian yang sudah pernah dilakukan, sehingga dapat bermanfaat dalam bidang kesehatan khususnya.

1.2 Rumusan Masalah

Tugas Akhir ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan metode *preprocessing Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA)?

2. Bagaimana menerapkan metode klasifikasi *Neural Network* pada data sinyal EEG?
3. Bagaimana menerapkan metode optimasi *Genetic Algorithm* pada metode klasifikasi *Neural Network*?
4. Bagaimana performa sistem yang dihasilkan?

1.3 Batasan Permasalahan

Permasalahan yang dibahas pada Tugas Akhir ini memiliki batasan sebagai berikut:

1. Dataset yang digunakan adalah data EEG recordings of control and alcoholic subjects dari *Neurodynamics Laboratory, State University of New York Health Center* yang terdiri dari 64 elektroda, 256 Hz.
2. Jumlah data yang digunakan adalah 260 data acak yaitu 130 alkoholik dan 130 normal.
3. Hasil klasifikasi data sinyal EEG terdiri dari 2 kelas, yaitu kelas pecandu alkohol dan kelas normal.
4. Perangkat lunak yang digunakan adalah Matlab R2014a.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Mengimplementasikan metode *Wavelet Packet Decomposition* (WPD) pada data EEG untuk mendekomposisi sinyal EEG dan ekstraksi fitur.
2. Mengimplementasikan metode *Principal Component Analysis* (PCA) untuk mereduksi jumlah dimensi data hasil ekstraksi fitur.
3. Mengimplementasikan kombinasi metode *Genetic Algorithm* dan *Neural Network* untuk menentukan hasil klasifikasi data.

1.5 Manfaat

Manfaat yang diperoleh dari pembuatan Tugas Akhir ini adalah dapat memberikan manfaat pada bidang kesehatan untuk membantu meningkatkan keakuratan hasil dalam melakukan diagnosis pasien yang termasuk pecandu alkohol atau normal. Hasil diagnosis nantinya dapat menentukan tindakan medis lanjutan bagi pasien pecandu alkohol.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahapan awal dari Tugas Akhir ini adalah penyusunan Proposal Tugas Akhir. Proposal Tugas Akhir berisi pendahuluan, deskripsi dan gagasan metode – metode yang dibuat dalam Tugas Akhir ini. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya Tugas Akhir, rumusan masalah yang diangkat, batasan masalah untuk Tugas Akhir, dan manfaat dari hasil pembuatan Tugas Akhir ini. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan Tugas Akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan Tugas Akhir.

1.6.2 Studi Literatur

Tugas Akhir ini menggunakan literatur *paper* yang berasal dari jurnal internasional bereputasi yaitu IEEE dan *Scioncedirect* untuk mencari informasi yang dapat dijadikan referensi dalam pengerjaan Tugas Akhir ini. Selain itu juga digunakan sejumlah referensi buku dan literatur lain yang berhubungan dengan klasifikasi dari data EEG dan metode – metode yang diusulkan pada Tugas Akhir ini termasuk tahap praproses berupa metode

Wavelet Packet Decomposition (WPD) Principal Component Analysis (PCA), dan kombinasi Genetic Algorithm dan Neural Network.

1.6.3 Implementasi Perangkat Lunak

Implementasi merupakan tahap untuk membangun metode-metode yang sudah diajukan pada proposal Tugas Akhir. Untuk membangun algoritma yang telah dirancang sebelumnya, maka dilakukan implementasi dengan menggunakan suatu perangkat lunak. Pada tugas akhir ini, perangkat lunak yang digunakan adalah Matlab R2014a.

1.6.4 Pengujian dan Evaluasi

Pada tahap ini metode dari perangkat lunak yang telah disusun dan diimplementasikan diuji coba dengan menggunakan data sinyal otak (EEG) yang sudah dikelola. Data EEG tersebut diuji coba menggunakan perangkat lunak yang telah diimplementasikan tersebut. Setelah itu hasil pengujian perangkat lunak dievaluasi dengan menghitung nilai akurasi, sensitifitas dan spesifitasnya.

1.6.5 Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

1. Bab I. Pendahuluan

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.

2. Bab II. Tinjauan Pustaka

Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan Tugas Akhir ini. Secara garis besar, bab ini berisi tentang alkoholisme, *Electroencephalographic* (EEG), *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan kombinasi *Genetic Algorithm* dan *Neural Network*.

3. Bab III. Perancangan Perangkat Lunak

Bab ini berisi pembahasan mengenai perancangan dari metode *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan kombinasi *Genetic Algorithm* dan *Neural Network*. yang digunakan untuk klasifikasi alkoholik dari data EEG.

4. Bab IV. Implementasi

Bab ini menjelaskan implementasi yang berbentuk kode sumber dari proses pengelolaan data EEG, metode *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan kombinasi metode *Genetic Algorithm* dan *Neural Network* untuk klasifikasi.

5. Bab V. Hasil Uji Coba dan Evaluasi

Bab ini berisikan hasil uji coba dari *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan kombinasi *Genetic Algorithm* dan *Neural Network* untuk klasifikasi alkoholik dan normal yang sudah diimplementasikan pada kode sumber.

6. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir, dan saran untuk pengembangan solusi ke depannya.

7. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam Tugas Akhir.

8. Lampiran

Dalam lampiran terdapat tabel-tabel data hasil uji coba dan kode sumber program secara keseluruhan.

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam Tugas Akhir. Teori-teori tersebut diantaranya adalah *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan kombinasi *Genetic Algorithm* dan *Neural Network* dan beberapa teori lain yang mendukung pembuatan Tugas Akhir.

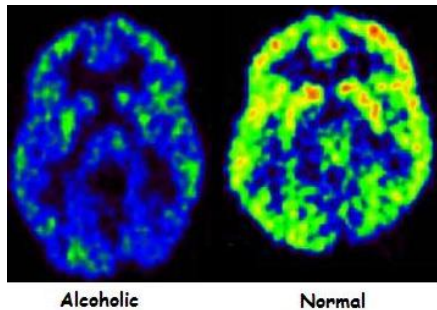
2.1 Alkoholisme

Alkoholisme adalah gangguan yang ditandai oleh konsumsi berlebihan dan ketergantungan pada alkohol. Alkoholisme merupakan salah satu penyakit kronis yang mencakup masalah-masalah termasuk mengendalikan pasien untuk minum, menjadi sibuk dengan alkohol, bahkan terus menggunakan alkohol sebagai solusi ketika sedang terjadi masalah, Pasien alkoholisme biasanya harus meminum alkohol lebih untuk mendapatkan efek yang sama (ketergantungan fisik). Hal tersebut disebabkan karena beberapa faktor yang melatarbelakangi seseorang untuk mengonsumsi alkohol sampai menjadi pecandu alkohol, seperti faktor genetika, biologis yang psikologis, lingkungan sosial, dan budaya.

Seseorang dapat dikatakan sebagai alkoholisme atau pecandu alkohol apabila ditandai dengan kondisi berikut:

- Meminum alkohol dalam jumlah besar selama periode waktu tertentu.
- Ketidakmampuan untuk mengontrol meminum alkohol.
- Mendapat dan meminum alkohol merupakan suatu hal yang menyenangkan
- Tidak dapat menahan untuk meminum alkohol
- Mengalami perubahan perilaku seperti gangguan dalam fungsi sosial dan kerja.

Pada bidang kesehatan, paparan kronis terhadap etanol, senyawa organik yang terdapat di alkohol akan merusak mitokondria hepatosit yang berpengaruh pada respirasi selular rantai pernapasan beserta respirasinya. Selain itu, efek toksis pada alkohol dapat menyebabkan kerusakan pada otak. Penelitian dari Carol Ann Paul, Wellesley College menunjukkan bahwa semakin seseorang banyak minum alkohol secara kronis, semakin otak akan berkurang volumenya. Hasil juga diadakan benar bahkan untuk peminum moderat (tidak terlalu sering minum/ sekali-kali minum alkohol). Gambar 2.1 merupakan perbandingan pasien pecandu alkohol dengan normal.



Gambar 2.1 Perbedaan Otak Alkoholik dan Normal [5]

2.2 Electroencephalography (EEG)

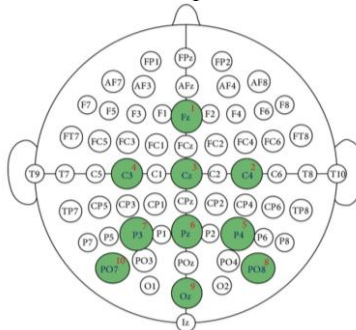
Electroencephalography (EEG) adalah prosedur pencatatan aktivitas listrik otak dengan alat pencatatan yang peka. EEG nantinya akan mengukur fluktuasi tegangan yang dihasilkan oleh arus ion di dalam neuron otak. Tujuan EEG berbasis *Brain Computer Interface* (BCI) adalah untuk mendeteksi pola sinyal EEG pengguna untuk mengontrol computer atau perangkat eksternal. Dalam konteks klinis, EEG mengacu kepada perekaman aktivitas elektrik spontan dari otak selama periode tertentu yang direkam dari banyak elektroda yang dipasang di kulit kepala. Neuron di korteks otak mengeluarkan gelombang-gelombang listrik dengan voltase yang sangat kecil (mV), kemudian dialirkan

ke mesin EEG untuk diamplifikasi sehingga terekamlah sinyal yang ukurannya cukup untuk dapat ditangkap oleh mata pembaca EEG. Pada orang normal, sinyal EEG dapat dibagi menjadi 4 grup berdasarkan frekuensinya. Jenis sinyal EEG tersebut dapat dilihat pada Tabel 2.1 [3]:

Tabel 2.1. Jenis Sinyal EEG

Nama Sinyal	Frekuensi
Delta	<3.5 Hz
Theta	4-7 Hz
Alpha	8-13 Hz
Beta	13-30 Hz

Pada bidang kesehatan masa kini, teknik modern seperti CT Scan dan *Magnetic Resonance Imaging* (MRI) dapat dapat secara langsung memeriksa otak, namun EEG tetap berguna mengingat sifatnya yang non-destruktif, dapat digunakan secara *online* dan dapat mengidentifikasi kondisi mental dan pikiran, serta menangkap persepsi seseorang terhadap rangsangan luar. Selain itu, tujuan digunakannya sinyal EEG untuk mendiagnosis penyakit yang berhubungan dengan kelainan pada otak, kejiwaan, gangguan tidur dan menganalisa penyakit alzheimer, epilepsi dan alkoholisme. Contoh dari konfigurasi EEG dapat dilihat pada Gambar 2.2 dan hasil rekaman EEG pada Gambar 2.3.



Gambar 2.2 Konfigurasi EEG [6]



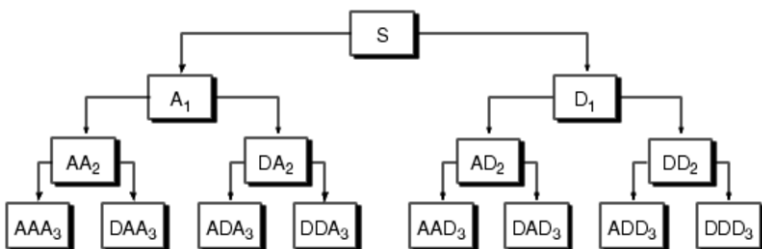
Wavelet Packet Decomposition merupakan metode turunan

Transformasi *wavelet* hampir menyerupai transformasi fourier yang merupakan bentuk pengklasifikasian sinyal menjadi bentuk sinusoidal dengan berbagai macam frekuensi. *Discrete Wavelet Transform* (DWT) merupakan salah bentuk diskrit dari *wavelet transform* dan seringkali digunakan. DWT terdiri dari

Transformasi *wavelet* hampir menyerupai transformasi fourier yang merupakan bentuk pengklasifikasian sinyal menjadi bentuk sinusoidal dengan berbagai macam frekuensi. *Discrete Wavelet Transform* (DWT) merupakan salah bentuk diskrit dari *wavelet transform* dan seringkali digunakan. DWT terdiri dari

proses sampling sinyal berdasarkan penskalaan dan pergeseran parameter. Proses DWT diawali dengan proses *filtering* dengan melewati *highpass filter* dan *lowpass filter*. Kemudian dilanjutkan dengan operasi sub-sampling dengan mengambil masing-masing setengah dari keluaran *filter*. Proses selanjutnya adalah modifikasi sinyal berdasarkan fungsi skala dan waktu, setelah itu, proses rekonstruksi sinyal pada tiap level koefisiennya. Seluruh proses ini dinamakan proses dekomposisi dan dapat dilakukan secara berlanjut hingga didapatkan tingkat dekomposisi yang diinginkan.

Sama halnya dengan DWT, Cara kerja *Wavelet Packet Decomposition* (WPD) dengan menggunakan wavelet yang di skala dan menerjemahkan ganda sifat asli dari wavelet utama yang disebut dengan ‘*mother wavelet*’ untuk mentransformasi sinyal inputan. Akan tetapi berbeda dengan DWT yang hanya mendekomposisi koefisien aproksimasinya saja, WPD akan tetap melakukan dekomposisi disetiap level koefisien detail dan aproksimasinya hingga membentuk *full binary tree* [8]. Untuk mendapatkan jumlah koefisien disetiap level dekomposisi n , WPD nantinya akan menghasilkan 2^n koefisien baru. Topologi metode WPD mendekomposisi 3 level dapat dilihat pada Gambar 2.3 dimana A menggambarkan koefisien aproksimasi dan D koefisien detail.



Gambar 2.4 Topologi WPD [9]

2.4 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah salah satu teknik reduksi dimensi yang merubah dataset dengan dimensi besar (dimensi - m) ke dimensi rendah ruang fitur ortogonal (dimensi - n , dimana $m > n$) akan tetapi tetap mempertahankan informasi dari dimensi besar dataset yang asli. Setiap fitur ortogonal yang dihasilkan disebut sebagai “*Principal Component*” atau PC. PCA memakai dekomposisi dengan *eigenvectors* dan *eigenvalues*. *eigenvectors* yang memiliki *eigenvalues* terbesar adalah vektor yang searah dengan bidang yang memiliki korelasi tertinggi.

Tahap pertama yang dilakukan dalam PCA adalah membuat matriks varians-kovarians pada diagonal yang mengandung nilai varians dan sisi lainnya mengandung nilai kovarians. Matriks varians-kovarians mempunyai dimensi $n \times n$ dimana n adalah jumlah atribut/dimensi dari data, X_i dan Y_i adalah data dan \bar{X} dan \bar{Y} adalah rata – rata dari masing – masing atribut data. Rumus mencari matriks varians-kovarians dapat dilihat pada persamaan 2.1 dan 2.2 [10].

$$var(x) = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)} \quad (2.1)$$

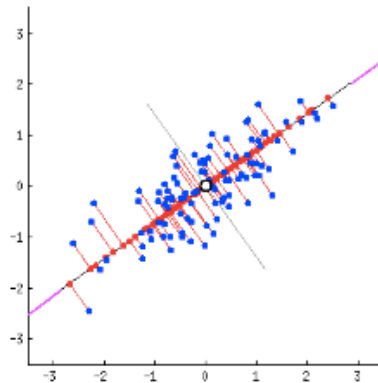
$$var(x) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (2.2)$$

Untuk mendapatkan kepentingan dari PC *eigenvalue* dan *eigenvector* sangat berperan penting. Oleh karena itu, tahapan selanjutnya dari PCA yaitu dengan mencari *eigenvalue* dan *eigenvector* dari setiap matriks kovarians. Untuk mencari nilai *eigenvalue* dan *eigenvector* dapat dilihat pada persamaan 2.3.

$$\Sigma v = \lambda v \quad (2.3)$$

dimana Σ adalah kovarians matriks, λ adalah *eigenvalues* dan v adalah *eigenvektors*.

Tahapan selanjutnya PC akan diurutkan berdasarkan tingkat kesesuaian *eigenvalue* yang mempresentasikan scalar dari tingkatan varian dalam PC yang sesuai. Oleh karena itu, PC yang paling pertama mempresentasikan varian dataset yang paling signifikan. Proses pereduksian dimensi dilakukan dengan mentransformasikan data awal ke set variabel yang baru (PC) yang tidak berkorelasi satu sama lain. Dalam kata lain, jumlah PC akan selalu lebih kecil atau sama dengan jumlah variabel awal [11]. Secara keseluruhan, topologi PCA dapat dilihat pada Gambar 2.5.



Gambar 2.5 Topologi PCA [10]

2.5 Kombinasi Genetic Algorithm dan Neural Network

Kombinasi *Genetic Algorithm* dan *Neural Network* adalah penggabungan metode jaringan saraf dan algoritma genetika. Penggunaan GA ditujukan untuk mencari inisial bobot dari suatu *neural network* dan mempercepat konvergensi. Hal ini digunakan karena algoritma *Back Propagation* membuat proses konvergensi menjadi lambat. Pada penelitian yang dilakukan sebelumnya, hasil penggabungan kedua algoritma diatas dapat meningkatkan kemampuan dari *Neural Network* dibandingkan dengan *Single Back Propagation*.

Genetic Algorithm sendiri digunakan untuk mengoptimasi bobot dan bias dari *neural network* yang diinisialisasi random diawal. GA nantinya akan mencari model *network* terbaik yang dihasilkan dari proses-proses penting GA. GA akan melakukan *generate* bobot dan bias awal menggunakan proses-proses penting dari GA dan menggunakan nilai error dari *Back Propagation Neural Network* untuk mengukur performa model *network* yang dihasilkan. *Network* dengan nilai error paling kecil merupakan model *network* terbaik yang nantinya akan digunakan untuk proses testing. Pada Tugas Akhir ini, nilai error tersebut menggunakan *Mean Square Error* (MSE) dari *network* tersebut. Perhitungan MSE dapat dilihat pada persamaan 2.5.

$$MSE = \frac{1}{n} \sum_{n=1}^n (a - y)^2 \quad (2.5)$$

dimana a merupakan kelas sesungguhnya dan y merupakan kelas prediksi [12].

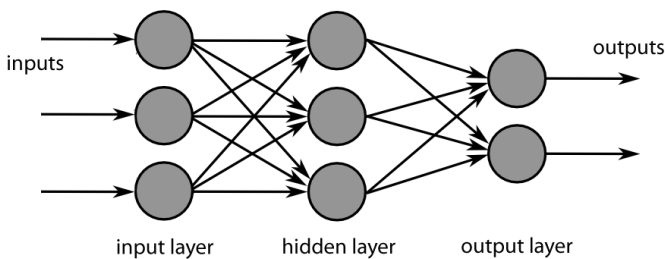
2.5.1 Back Propagation Neural Network

Back Propagation Neural Network (BPNN) merupakan algoritma pelatihan terbimbing yang mempunyai banyak lapisan. BPNN menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Terdapat 3 fase dalam pelatihan BPNN, yaitu fase maju (*feed forward*), fase mundur (*backpropagation*), dan fase modifikasi bobot. Rumus untuk perhitungan bobot :

$$w_{lji}(k+1) = w_{lji}(k) - \mu \frac{\partial MSE(w)}{\partial w_{lji}} \quad (2.6)$$

dimana w_{lji} adalah bobot antara neuron i pada layer $l-1$ dan neuron j pada layer l , μ adalah bilangan real positif, *learning rate* yang berfungsi untuk mengontrol step pembelajaran algoritma dan

biasanya diset kecil [13]. Dalam fase *back propagation*, tiap-tiap unit output menerima target pola yang berhubungan dengan pola input untuk dihitung nilai kesalahan. Kesalahan tersebut akan dipropagasikan mundur. Sedangkan fase modifikasi bobot bertujuan untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang secara terus menerus hingga kondisi penghentian dipenuhi. Secara garis besar, Arsitektur *Back Propagation Neural Network* ditunjukkan pada Gambar 2.6.



Gambar 2.6 Ilustrasi Back Propagation Neural Network

2.5.2 Genetic Algorithm (GA)

Penggunaan *Genetic Algorithm* (GA) sebagai penyelesaian masalah sudah bukan hal yang baru. Pada tahun 1970, John H. Holland terbukti menjadi kontribusi yang signifikan untuk aplikasi ilmiah dan rekayasa GA. Sejak saat itu, output pekerjaan penelitian di bidang ini telah berkembang pesat mengenai penelitian ini diseluruh institusi akademik di seluruh dunia [14]. GA sendiri adalah salah satu jenis metode pencarian stokastik global dan metode optimasi yang mengadaptasi proses seleksi alam yang meniru natural evolusi biologi. GA beroperasi dimulai dari populasi solusi yang potensial. Populasi adalah kumpulan individu / *chromosome*, yang tiap individu / *chromosome* nya merepresentasikan solusi potensial yang optimal dari suatu permasalahan. Pada tiap generasi yang dihasilkan GA, akan dibuat set set solusi kemungkinan baru yang didapatkan dari memilih

individu – individu terbaik, dilihat dari level *fitness* nya. GA memiliki beberapa operasi penting yaitu:

- Inisialisasi

Ukuran populasi yang akan diproses selanjutnya akan sangat tergantung pada proses Inisialisasi, tetapi biasanya proses inisialisasi berisi beberapa ratusan atau ribuan yang mungkin terjadi pada satu proses optimasi. Tujuan dari inisialisasi sendiri adalah untuk melakukan populasi awal secara acak, yang memungkinkan seluruh solusi terbaik terpilih. Hal ini disebabkan karena solusi yang paling unggul terkadang berada di daerah di mana solusi optimal mungkin ditemukan.

- Seleksi

Seleksi adalah operasi memilih individu baru sebagai orang tua (*parent*), maksudnya adalah semakin tinggi nilai *fitness* suatu individu maka makin tinggi pula kemungkinannya untuk dapat muncul pada generasi selanjutnya. Ada berbagai jenis metode seleksi pada *Genetic Algorithm*, diantaranya adalah *Roulette Wheel Selection* (RWS), *Stochastic Universal Sampling* (SUS), *Linear Rank Selection* (LRS), *Exponential Rank Selection* (ERS), *Tournament Selection* (TOS), dan *Truncation Selection*. Pada Tugas Akhir ini, proses seleksi dari GA menggunakan *Roulette Wheel Selection* (RWS).

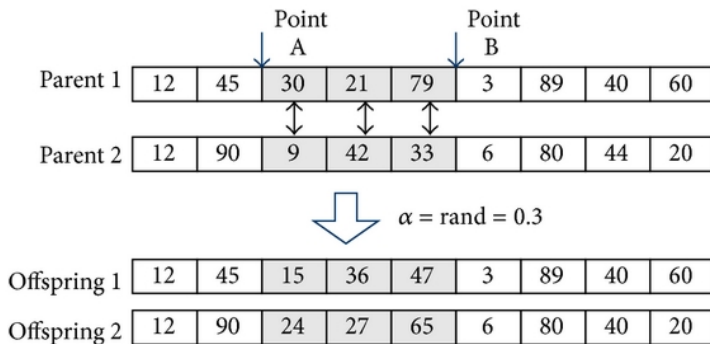
- Crossover

Crossover merupakan operasi saling bertukar beberapa bagian pada 2 individu *parent*, dimulai dari *crossover point* yang ditentukan. Operasi ini dilakukan untuk mengaktifkan proses evolusi menuju ruang hasil yang lebih baik. Apabila nilai dari kromosom merupakan bilangan asli, *crossover* dianjurkan untuk menggunakan teknik *Aritmatic Crossover*. Ada beberapa jenis *Aritmetic Crossover*, diantaranya adalah

single, *simple* dan *whole*. Pada Tugas Akhir ini digunakan *simple arithmetic crossover* yang menukar beberapa gen dari titik – titik yang ditentukan, misal dari gen awal sampai *crossover point* tertentu atau bisa juga dari *crossover point* sampai gen terakhir. Berikut adalah rumus dari *arithmetic crossover*:

$$\begin{cases} X_A^{i+1} = \alpha X'_B + (1 - \alpha)X'_A \\ X_B^{i+1} = \alpha X'_A + (1 - \alpha)X'_B \end{cases} \quad (2.7)$$

dimana alpha (α) adalah parameter yang dapat dipilih secara konstan / variabel yang ditentukan oleh generasi evolusi [13]. Ilustrasi dari *arithmetic crossover* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Ilustrasi Arithmetic Crossover

- **Mutasi**

Mutasi adalah proses mengganti beberapa nilai gen pada chromosome, tujuannya adalah untuk menghindari *local convergence* pada GA. Mutasi diperlukan untuk mengembalikan informasi bit yang hilang akibat dari *crossover*. Mutasi diterapkan dengan probabilitas yang sangat kecil. Jika mutasi dilakukan terlalu sering, maka akan menghasilkan individu yang lemah karena konfigurasi gen

pada individu yang unggul akan dirusak. Berdasarkan bagian yang termutasi, proses mutasi dapat dibedakan jadi beberapa bagian. Salah satunya mutasi 1 bit saja yang berubah. Pada Tugas Akhir ini, operator mutasi yang digunakan adalah *uniform mutation* dengan mengganti nilai kromosom terpilih dengan nilai acak antara nilai minimum dan maximum dari kromosom *parent*. Ilustrasi pada mutasi dapat dilihat pada Gambar 2.8.

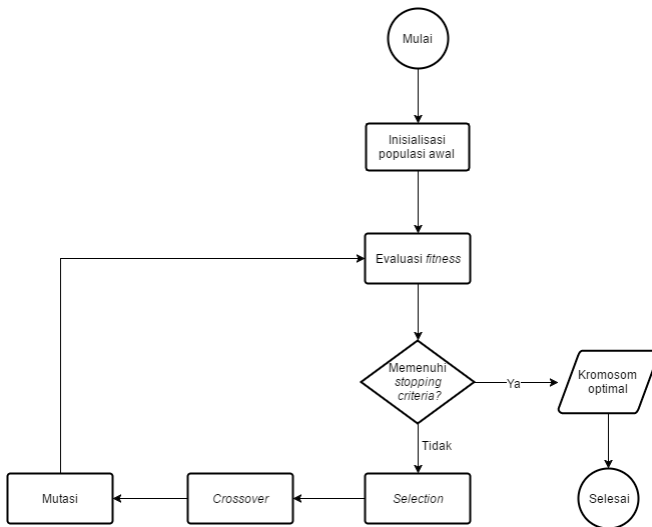
Awal	0	1	1	0	1	1	0	0
Akhir	0	1	0	0	1	1	0	0

Gambar 2.8 Ilustrasi Mutasi

- **Evaluasi**
Evaluasi pada GA adalah saat menghitung performa/ *fitness value* pada suatu individu/ kromosom untuk mendapatkan *fitness value* pada GA macam-macam, tergantung pada permasalahan yang akan diselesaikan. Pada Tugas Akhir ini, perhitungan *fitness value* menggunakan nilai *Mean Square Error* (MSE) pada persamaan 2.5 dari metode *Back Propagation Neural Network* dari individu yang terbentuk sebelumnya.
- **Stopping Criteria**
Proses operasi terakhir dari *Genetic Algorithm* yang telah dijelaskan diatas terus dilakukan hingga memenuhi *Stopping Criteria*. *Stopping Criteria* adalah kriteria yang digunakan untuk menghentikan proses jalannya GA. Hal tersebut bertujuan untuk mencari generasi terbaik dari populasi yang sudah dihasilkan dari proses penting GA. Terdapat berbagai kriteria yang dapat digunakan untuk menghentikan jalannya GA, yaitu:

- Telah tercapainya generasi yang diinginkan.
- Dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi yang tidak berubah lagi.
- Memenuhi kondisi tertentu yang mempertimbangkan kondisi atau waktu tertentu.
- Tidak ditemukan nilai *fitness* yang lebih baik dari sebelumnya.
- Gabungan dari kondisi diatas.

Seperti yang telah dijelaskan pada sub bab sebelumnya, GA akan selalu membuat generasi-generasi baru untuk mendapatkan global optimum setelah melewati proses – proses penting GA. Operasi – operasi dari GA ini harus bersifat sederhana, berguna, dan tahan dalam menghasilkan generasi-generasi baru. Hal ini disebabkan karena GA mengabungkan kromosom – kromosom dan menghasilkan generasi baru dengan mempertimbangkan peluang yang efektif dan efisien. Secara garis besar, diagram alir dari proses GA dapat dilihat pada Gambar 2.9.



Gambar 2.9 Diagram alir proses GA

2.6 Confusion Matrix

Confusion matrix adalah suatu metode yang biasa digunakan untuk melakukan perhitungan akurasi pada konsep data mining. *Confusion Matrix* memiliki informasi hasil prediksi dan aktual pada data yang telah di klasifikasi. Performa suatu sistem biasanya dievaluasi menggunakan *Confusion Matrix*. Ciri dari *Confusion Matrix* sendiri adalah kolom mempresentasikan kelas prediksi, sedangkan baris mempresentasikan kelas sebenarnya atau sebaliknya. Tabel 2.2 menunjukkan *Confusion Matrix* pada dua kelas.

Tabel 2.2 Confusion Matrix dua kelas

		PREDIKSI	
		1	0
AKTUAL	1	TP	FN
	0	FP	TN

Nilai yang bisa dihitung menggunakan *Confusion Matrix* yaitu akurasi, sensitivitas dan sensitivitas. Akurasi merupakan hasil bagi dari jumlah prediksi yang terklasifikasi secara benar dibagi total data yang diklasifikasi seperti pada persamaan 2.8.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.8)$$

Sensitivitas adalah perbandingan dari jumlah data TP dengan total data TP dan FN seperti pada persamaan 2.9.

$$Sensitivitas = \frac{TP}{TP+FN} \quad (2.9)$$

Spesifisitas adalah perbandingan dari jumlah data TN dengan total data TN dan FP seperti pada persamaan 2.10.

$$Spesifisitas = \frac{TN}{TN+FP} \quad (2.10)$$

(Halaman ini sengaja dikosongkan)

BAB III

PERANCANGAN SISTEM

Bab ini menjelaskan tentang perancangan dan pembuatan sistem perangkat lunak. Sistem perangkat lunak yang dibuat pada Tugas Akhir ini adalah mengolah data EEG dengan metode *Wavelet Packet Decomposition* (WPD) sebagai *preprocessing* awal yaitu mendekomposisi sinyal yang selanjutnya diekstraksi fitur sebagai ciri dari data masukan, *Principal Component Analysis* (PCA) untuk mereduksi dimensi data hasil ekstraksi fitur dan klasifikasi data menggunakan *Back Propagation Neural Network* yang dioptimasi menggunakan metode *Genetic Algorithm* (GA). Pada bab ini pula akan dijelaskan gambaran umum sistem dalam bentuk *flowchart*.

3.1 Data

Pada sub bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan perangkat lunak untuk selanjutnya diolah dan dilakukan pengujian sehingga menghasilkan data keluaran yang diharapkan.

3.1.1 Data Masukan

Data masukan adalah data yang digunakan sebagai masukan awal dari sistem. Data yang digunakan dalam perangkat lunak Klasifikasi Alkoholik menggunakan *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan kombinasi *Genetic Algorithm* dan *Neural Network* adalah data sinyal otak (EEG) yang diunduh dari *website UCI Machine Learning Repository*. Dataset EEG yang digunakan adalah versi *The Large Dataset* yang terdiri dari data training yang memiliki data 10 orang alkoholik dan 10 orang normal yang direkam sebanyak 30 kali. Pada data testing, rekaman didapat dari orang yang sama namun rekaman nya berbeda, sehingga terdapat 600 data *training* dan 600

data *testing*. Pada setiap data terdapat 64 *channel* dengan frekuensi 256 Hz per detik. Pemilik asli data adalah dari *Neurodynamics Laboratory, State University of New York Health Centre*. Pada Tugas Akhir ini akan diambil 260 data acak yaitu 130 alkoholik dan 130 normal yang akan dibagi menjadi 2 dataset. Dataset pertama terdiri dari 60 data *training* dan 40 data *testing*, sedangkan dataset kedua terdiri 120 data *training* dan 40 data *testing*. Data ini selanjutnya akan diproses pada Tugas Akhir ini. Contoh data awal yang digunakan pada Tugas Akhir ini terdapat pada Gambar 3.1.

```

1 # co2a0000364.rd
2 # 120 trials, 64 chans, 416 samples 368 post_stim samples
3 # 3.906000 msec uV
4 # S1 obj , trial 30
5 # FP1 chan 0
6 30 FP1 0 -3.550
7 30 FP1 1 -5.015
8 30 FP1 2 -5.503
9 30 FP1 3 -3.550
10 30 FP1 4 -0.621
11 30 FP1 5 1.821
12 30 FP1 6 2.309
13 30 FP1 7 0.844
14 30 FP1 8 0.844
15 30 FP1 9 3.286
16 30 FP1 10 7.680
17 30 FP1 11 10.122
18 30 FP1 12 9.145
19 30 FP1 13 5.239
20 30 FP1 14 0.844
21 30 FP1 15 -0.621

```

Gambar 3.1 Contoh Data Awalan

Seperti yang dapat dilihat pada Gambar 3.1, baris pertama adalah nama file dari data tersebut, baris ke dua sampai ke empat adalah deskripsi dari data, sedangkan baris ke lima adalah nama *channel*. Dimulai dari baris ke enam data diinterpretasikan dengan beberapa kolom yaitu kolom pertama dari data awalan merupakan nomor percobaan, kolom ke kedua merupakan nama *channel*, kolom ketiga merupakan nomor data dalam *channel*, sedangkan kolom keempat adalah nilai dari sinyal EEG.

3.1.2 Data Keluaran

Data masukan yang telah dibagi menjadi dua data yaitu data pembelajaran atau *training* dan data pengujian atau *testing* akan diproses menggunakan *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan Kombinasi *Genetic Algorithm* dan *Neural Network*. Selanjutnya, program yang dikembangkan dapat mengeluarkan hasil dari proses klasifikasi data testing berupa hasil akurasi, sensitivitas, dan spesifitas.

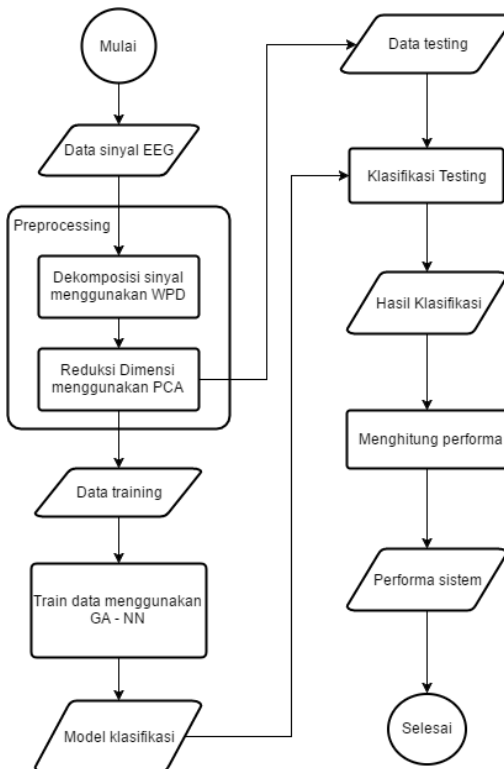
3.2 Desain Umum Sistem

Rancangan bangun perangkat lunak yang akan dibuat untuk Klasifikasi Data EEG Alkoholik Menggunakan Metode *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan kombinasi *Genetic Algorithm* dan *Neural Network*. Proses pertama yang dilakukan sistem yaitu mendekomposisi data sinyal inputan menggunakan *Wavelet Packet Decomposition* (WPD), selain itu WPD juga bertujuan untuk ekstraksi fitur data inputan dengan mengambil masing-masing nilai yang mempresentasikan data tersebut dari masing-masing level.

Tahapan selanjutnya dari praproses sebelum proses klasifikasi yaitu mereduksi dimensi data menggunakan *Principal Component Analysis* (PCA). Tujuan utama dari PCA yaitu untuk memperingan kinerja mesin klasifikasi terutama dalam hal waktu komputasi. Cara kerja PCA yaitu mentransformasi fitur ke dimensi lain dengan menghitung tingkat kepentingan informasi dari data yang akan diambil berdasarkan nilai *eigen value* data tersebut.

Hasil dari PCA yaitu data hasil ekstraksi fitur dengan dimensi data yang baru. Data tersebut selanjutnya dianggap sudah siap untuk masuk ke proses klasifikasi menggunakan metode *Neural Network* yang dioptimasi menggunakan *Genetic Algorithm* dalam pembaharuan bobot dan bias awal *network*. Seperti yang dijelaskan sebelumnya, hasil klasifikasi dari sistem Klasifikasi

Data EEG Alkoholik menggunakan *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan Kombinasi *Genetic Algorithm* dan *Neural Network* yaitu perhitungan performa berupa akurasi, sensitivitas, dan spesitifitas dari data testing. Secara umum, proses dari sistem Klasifikasi EEG Alkoholik menggunakan *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan kombinasi *Genetic Algorithm* dan *Neural Network* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alir Keseluruhan Sistem

Preprocessing data atau bisa disebut juga dengan data praproses adalah suatu proses/langkah yang dilakukan untuk membuat data mentah menjadi data yang berkualitas (input yang baik untuk proses data mining). Secara garis besar, proses *preprocessing* pada Tugas Akhir ini menggunakan metode *Wavelet Packet Decomposition* (WPD) untuk dekomposisi dan ekstraksi fitur sinyal, serta *Principal Component Analysis* (PCA) dalam reduksi dimensi data fitur.

3.2.1 Wavelet Packet Decomposition (WPD)

Wavelet Packet Decomposition (WPD) merupakan metode yang digunakan untuk mendekomposisi sinyal, tetapi tetap mempertahankan nilai dari aproksimasi dan detail dari sinyal tersebut. Sebelum masuk ke proses WPD, data awalan diproses terlebih dahulu untuk dapat masuk ke proses dekomposisi sinyal.

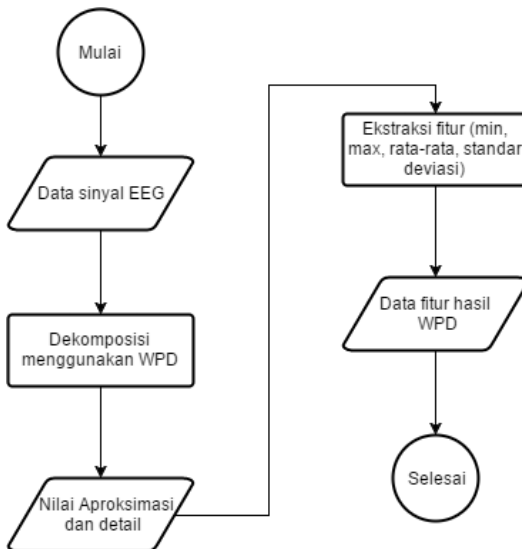
Pada Tugas Akhir ini, proses pertama yang dilakukan adalah mengelola data mentah yang diunduh dari *website* UCI *Machine Learning Repository* menjadi data yang dapat dikelola pada sistem Klasifikasi Data EEG Alkoholik menggunakan *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan Kombinasi Metode *Genetic Algorithm* dan *Neural Network*. Contoh data mentah yang digunakan pada tugas akhir ini dapat dilihat pada Gambar 3.1.

Proses pertama yang dilakukan pada Tugas Akhir ini yaitu mengambil nilai dari kolom ke 4 dan menghapus baris yang berawalan #. Hasil dari pemrosesan tersebut adalah matriks dengan ukuran 16384 x 1 yang terdiri dari 64 *channel* untuk satu data sinyal. Selanjutnya dilakukan penghapusan 4 *channel* terakhir sehingga satu data sinyal menjadi matriks baru dengan ukuran 15360 x 1. Hal tersebut dikarenakan 4 *channel* terakhir merupakan *channel* pendukung saat pengambilan data EEG, sehingga perlu adanya penghapusan dari *channel* tersebut.

Setelah dilakukan proses pengubahan bentuk data sinyal seperti penjelasan di atas, data masukan yang baru kemudian

digunakan sebagai input dari proses WPD. WPD sendiri bertujuan untuk memilah milih suatu sinyal data berdasarkan karakteristik yang menonjol dari sinyal tersebut. Penjelasan lebih lanjut tentang WPD dapat dilihat pada bab 2.3. Proses WPD menggunakan tipe *Wavelet Daubechies* 4 yang didekomposisi sebanyak 3 level.

Hasil dekomposisi sinyal 3 level menggunakan WPD adalah 14 buah koefisien yang terdiri dari aproksimasi dan detail. Data tersebut kemudian masuk ke tahapan ekstraksi fitur dengan mengambil nilai standar deviasi, minimum, maximum, dan rata-rata dari masing-masing koefisien yang sudah didekomposisi. Selanjutnya nilai – nilai hasil ekstraksi fitur tersebut disimpan dalam suatu matriks hingga menghasilkan 56 fitur baru untuk data training maupun data testing. Secara garis besar, proses WPD dapat dilihat pada Gambar 3.3.



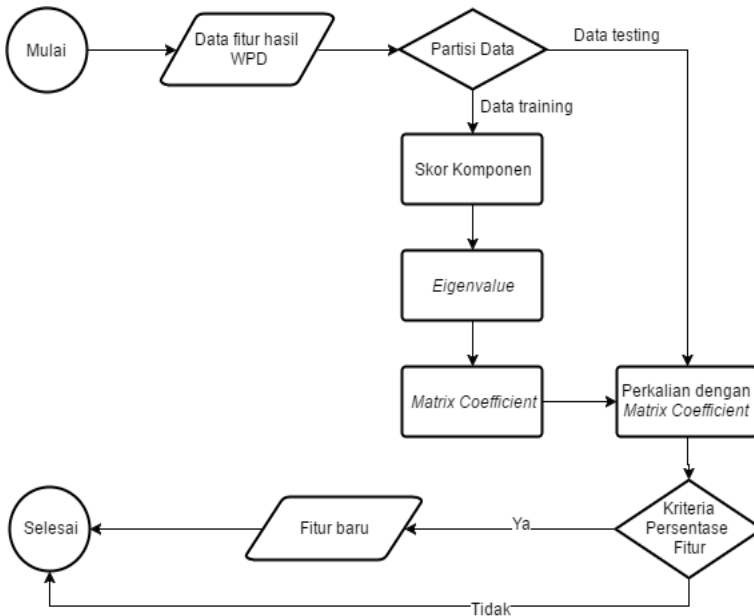
Gambar 3.3 Langkah Dekomposisi Sinyal WPD

3.2.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) pada Tugas Akhir ini bertujuan untuk mereduksi dimensi dari data hasil ekstraksi fitur menggunakan WPD yang sudah diproses sebelumnya. Selain itu, reduksi dimensi juga bertujuan untuk memperingan kinerja proses klasifikasi dan mengambil informasi yang relevan agar tidak mempengaruhi validasi data. Prinsip kerja PCA adalah dengan mentransformasikan fitur ke dimensi lain dengan menghitung tingkat kepentingan informasi yang diambil berdasarkan *Eigen Value* dari data tersebut.

Data input dari PCA adalah data hasil ekstraksi fitur WPD yang terdiri dari 56 fitur. Karena tujuan dari PCA adalah reduksi dimensi data fitur, maka output dari proses reduksi dimensi adalah terdapat fitur-fitur baru dengan dimensi yang lebih rendah namun tetap mempunyai kepentingan yang sama. Sebelum memasuki proses PCA, data training dan data testing harus dipartisi terlebih dahulu. Hal ini disebabkan karena proses reduksi dimensi data testing merupakan proyeksi data testing dengan nilai loading (COEFF) yang didapatkan dari data training. Oleh sebab itu, proses PCA dari data training harus dikerjakan terlebih dahulu sebelum proses PCA dari data testing.

Hasil reduksi dimensi PCA dari data training menghasilkan output yang terdiri dari nilai loading (COEFF), nilai hasil skor komponen (SCORE) dan nilai eigen (LATENT). Setelah proses PCA data training dilakukan, selanjutnya adalah proses PCA dari data testing yaitu perkalian dari data testing dengan nilai loading (COEFF) dari data testing. Hasil perkalian ini merupakan hasil reduksi dimensi PCA dari data testing. Hasil reduksi dimensi PCA tersebut kemudian disimpan di matriks yang berbeda antara data training dan data testing berdasarkan jumlah fitur yang nantinya akan masuk pada proses uji coba. Pengambilan fitur ini juga dengan memprioritaskan nilai eigen (LATENT) yang dihasilkan dari data training. Secara garis besar, proses PCA pada Tugas Akhir ini dapat dilihat pada Gambar 3.4.



Gambar 3.4 Langkah Reduksi Dimensi PCA

3.3 Kombinasi Genetic Algorithm dan Neural Network

Seperti yang telah dijelaskan pada bab sebelumnya, kombinasi metode *Genetic Algorithm* dan *Neural Network* merupakan algoritma klasifikasi *Neural Network* yang digabungkan dengan algoritma optimasi *Genetic Algorithm*. Pada Tugas Akhir ini, implementasi *Neural Network* terdiri dari 3 layer, yaitu *input layer*, *hidden layer*, dan *output layer*.

Pada proses *backpropagation* ada dua hal penting yang perlu diperhatikan, yaitu *learning rate* dan update bobot. *Learning rate* menentukan seberapa cepat jaringan saraf tersebut mempelajari pola dari data training. Nilai ini harus dipilih dengan benar, jika terlalu kecil maka proses pembelajaran akan lama dan jika terlalu besar maka akan terjadi penyimpangan. Sedangkan

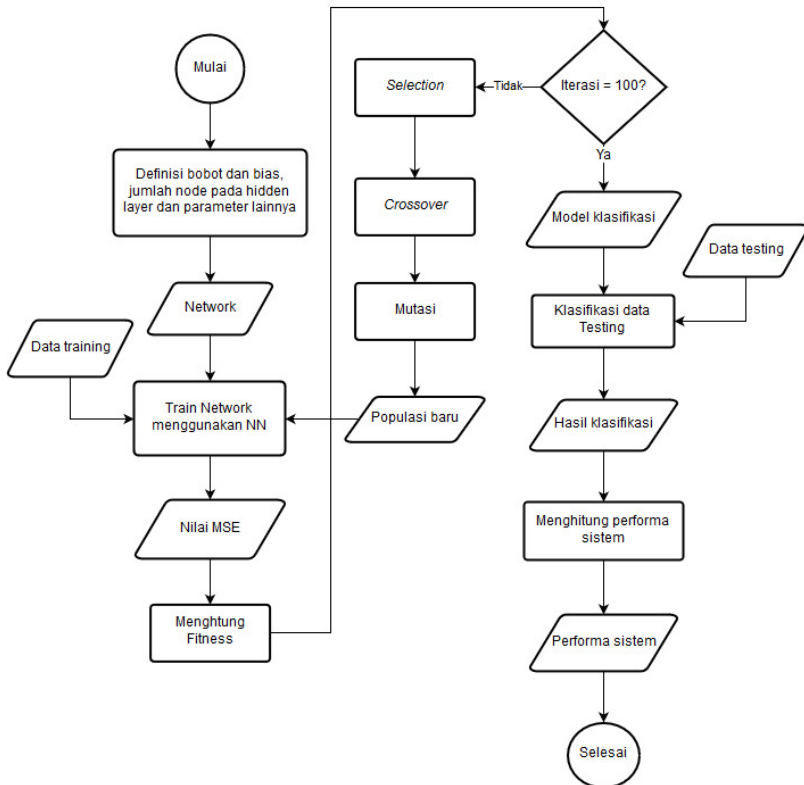
update bobot menunjukkan bahwa bobot pada iterasi sebelumnya mempengaruhi bobot sekarang. Oleh karena itu, pada Tugas Akhir ini *Genetic Algorithm* digunakan untuk mencari bobot dan bias terbaik yang dihasilkan oleh suatu *Neural Network*.

Selain itu, untuk training jaringan saraf perlu dimasukkan sejumlah inputan data yang kemudian akan menghasilkan beberapa output. Untuk membandingkan hasil output ini dengan target output setiap inputan, diperlukan perhitungan nilai error. Perhitungan nilai error inilah yang dapat menentukan bahwa hasil output benar atau salah. Selain itu, nilai error dari jaringan saraf ini digunakan untuk melakukan pembaharuan bobot dan bias dari *neural network* menggunakan GA. GA akan tetap melakukan *generate* bobot dan bias menggunakan proses penting GA yang terdiri dari seleksi, *crossover*, mutasi, dan evaluasi hingga memenuhi kondisi pemberhentian. Terdapat dua evaluasi dari GA pada Tugas Akhir ini, yang pertama ketika mencari bobot dan bias untuk populasi awal GA dan yang kedua ketika memasang bobot dan bias hasil dari generasi GA. Pada Tugas Akhir ini, perhitungan error tersebut menggunakan *Mean Square Error* (MSE). Secara garis besar, alur kombinasi metode *Genetic Algorithm* dan *Neural Network* dapat dilihat pada Gambar 3.5.

3.3.1 Pembangunan dan Pengaturan Network

Sebelum dapat digunakan, *network* pada *Neural Network* harus dibangun terlebih dahulu dengan mengatur parameter-parameter dan fungsi aktivasinya terlebih dahulu. Pengaturan tersebut diantaranya:

- Jumlah node *input layer*
Jumlah node *input layer* pada Tugas Akhir ini tergantung pada jumlah fitur yang dihasilkan PCA yang akan masuk pada skenario uji coba.
- Jumlah *hidden layer*
Jumlah *hidden layer* pada Tugas Akhir ini adalah 1 layer.



Gambar 3.5 Diagram Alir Klasifikasi GA-NN

- Jumlah node pada *hidden layer*
Jumlah node *hidden layer* pada Tugas Akhir ini adalah 15 node.
- Jumlah node *output layer*
Jumlah node pada *output layer* pada Tugas Akhir ini sebanyak jumlah kelas yang dihasilkan yaitu 2 node.
- *Learning rate*
Pada Tugas Akhir ini, *Learning rate* (μ) akan masuk pada skenario uji coba dengan mengganti nilai dari *Learning rate* (μ) yang diinisialisasi di awal.

- **Momentum**
Tujuan dari penerapan momentum agar suatu *network* merespon tidak hanya pada gradien lokal tetapi juga dapat merespon terhadap perubahan *trend* pada kurva error ketika merubah bobot. Pada Tugas Akhir ini, parameter momentum diinisialisasi di awal yaitu 0.35.
- **Fungsi aktivasi**
Jenis fungsi aktivasi pada Tugas Akhir ini menggunakan aktivasi log sigmoid dengan seperti yang ditunjukkan pada persamaan 3.1 rumus sebagai berikut:

$$\text{sigmoid}(x_i) = \frac{1}{1+\exp(-x_i)} \quad (3.1)$$

3.3.2 Pengaturan Parameter Awal Genetic Algorithm

Sama seperti halnya *Neural Network*, metode GA juga memiliki beberapa parameter yang harus diatur terlebih dahulu. Pengaturan parameter pada GA adalah sebagai berikut:

- **Representasi Kromosom**
Kromosom pada GA mewakili nilai bobot dan bias awal dari *Neural Network*. Konfigurasi bobot dan bias pada satu kromosom ditunjukkan pada Gambar 3.6 dimana n merupakan node pada *layer* awal dan m merupakan node pada *layer* tujuan.

W1			B1			W2			B2		
Wnm	Wnm	Bnm	Bnm	Wnm	Wnm	Bnm	Bnm

Gambar 3.6 Representasi Kromosom

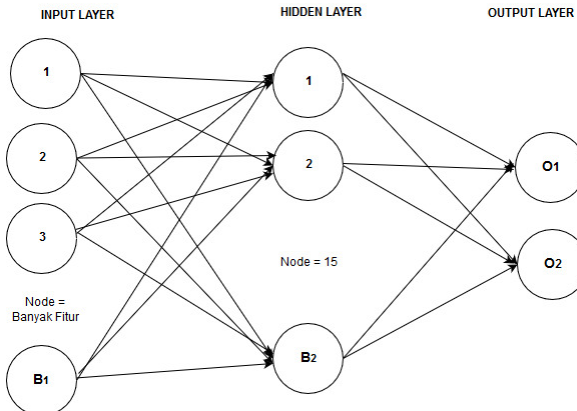
- **Jumlah populasi awal**
Pada Tugas Akhir ini, jumlah populasi awal pada GA adalah 15 populasi yang dianggap cukup untuk mempresentasikan populasi dari masing-masing bobot dan bias.
- **Jumlah generasi yang dihasilkan**

Dengan mempertimbangkan kompleksitas dan waktu komputasi, pada Tugas Akhir ini jumlah generasi yang nantinya akan dihasilkan oleh GA adalah 100 generasi.

- *Alpha (α)* pada proses *crossover*
Pada Tugas Akhir ini, nilai *alpha* pada proses *crossover* akan masuk pada skenario uji coba dengan mengganti nilai dari *Alpha (α)* yang diinisialisasi di awal.
- *Mutation Probability* pada proses mutasi
Konstanta *Mutation Probability* dari GA akan diinisialisasi diawal yaitu 0.095.
- *Fitness Function*
Perhitungan *fitness* pada Tugas Akhir ini menggunakan MSE pada *Neural Network* dari individu yang terbentuk. Perhitungan *fitness* ditunjukkan pada persamaan 3.2:

$$f = \frac{1}{MSE} \quad (3.2)$$

Secara garis besar, ilustrasi pembuatan *network* pada Tugas Akhir ini dapat dilihat pada Gambar 3.7.



Gambar 3.7 Ilustrasi Pembuatan Network

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa kode sumber untuk membangun program. Sebelum masuk ke penjelasan implementasi, akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Implementasi Klasifikasi Data EEG Alkoholik menggunakan *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan Kombinasi *Genetic Algorithm* dan *Neural Network* menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Spesifikasi lingkungan implementasi

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz 2.39 GHz
	Memori	8 GB 1600
Perangkat Lunak	Sistem Operasi	Windows 10 Pro
	Perangkat Pengembang	MATLAB R2014a

4.2 Implementasi

Pada sub bab implementasi akan menjelaskan bagaimana pembangunan perangkat lunak secara detail dan menampilkan kode sumber yang digunakan mulai dari tahap *preprocessing* sampai tahap klasifikasi.

4.2.1 Implementasi Pemrosesan Data Sebelum Praprocessing

Seperti yang telah dijelaskan pada bab perancangan sistem, bahwa hal yang pertama dilakukan sebelum membangun sistem Klasifikasi Data EEG Alkoholik menggunakan *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan Kombinasi *Genetic Algorithm* dan *Neural Network* adalah mengambil data pada kolom ke empat pada dataset mentah yang diunduh dari *UCI Machine Learning Repository*. Hal tersebut dikarenakan data pada kolom ke empat merupakan nilai dari sinyal tersebut dan dibutuhkan untuk mengklasifikasikan data EEG.

Hal pertama yang dilakukan pada Tugas Akhir ini adalah menghapus kolom pertama sampai ketiga serta baris yang berawalan # dari data mentah, setelah itu mengubah file berformat .rd menjadi file berformat .txt (contoh dari data mentah dapat dilihat pada Gambar 3.1). Hal tersebut bertujuan agar panjang sinyal per *channel* tidak berubah, karena setiap baris yang memulai *channel* baru selalu diawali # yang berisi keterangan dari *channel* tersebut. Implementasi dari merubah raw data mentah berformat .rd menjadi data berformat .txt dapat dilihat pada kode sumber 4.1 dengan menggunakan bahasa pemrograman python.

```

1  with open(fileInput) as f:
2      lines = f.readlines()
3
4      fileOutput = "a" + str(i) + ".txt"
5      target = open(fileOutput, 'w')
6
7      for isi in lines:
8          if isi[0] != '#':
9              coba = isi.split(' ')
10             target.write(coba[3])
11         target.close()

```

Kode Sumber 4.1 Pemrosesan Data Awal

Pada Kode Sumber 4.1 hal pertama yang dilakukan adalah membaca isi file yang telah diinisialisasi pada *directory*. Selanjutnya membuat file baru dengan format .txt seperti yang ditunjukkan kode sumber pada baris ke 4 dan 5. Pada baris ke 7-11 bertujuan untuk menghapus baris pada data yang berawalan # dan hanya mengambil data pada kolom ke 4.

4.2.2 Implementasi Wavelet Packet Decomposition (WPD)

Hasil dari proses data awalan seperti yang telah dijelaskan diatas adalah data sinyal dengan panjang baris 16384 yang telah disimpan dalam file berformat .txt. Hal yang dilakukan selanjutnya adalah menghapus 4 *channel* terakhir. Hal tersebut dikarenakan 4 *channel* terakhir merupakan *channel* pendukung saat pengambilan data, sehingga perlu adanya penghapusan dari *channel* tersebut. Implementasi pemotongan sinyal tersebut dapat dilihat pada Kode Sumber 4.2

1	%pengambilan 60 channel pertama
2	for i = 1:length(training)
3	tmp =
4	strcat('D:\TA\DATASET\training\ ', training(i).name);
5	tmp1 = load(tmp);
6	tmp2 = tmp1;
7	data = tmp2(1:15360);
8	
9	%mengambil nama file data yang di load utk disimpan
10	kembali dengan nama file yang sama
11	outfile = training(i).name;

Kode Sumber 4.2 Kode program penghapusan sinyal

Pada Kode Sumber 4.2, kode program pada baris ke 1 sampai 7 bertujuan untuk mengambil hanya 15360 baris pertama saja dari data sinyal atau 60 *channel* pertama. Pada baris ke 11 bertujuan untuk menyimpan nama file yang sedang di proses dan nantinya akan menyimpan hasil dari proses diatas dengan nama file yang sama.

Hasil dari Kode Sumber 4.2 adalah sinyal EEG yang sudah siap untuk diproses ke tahapan dekomposisi sinyal menggunakan WPD. Pada Tugas Akhir ini, sinyal akan didekomposisi menggunakan fungsi *Daubechies*. Selanjutnya hasil dari dekomposisi sinyal WPD akan menghasilkan koefisien-koefisien baru yang terbentuk seperti *full binary tree*. Penjelasan lebih lanjut tentang WPD dapat dilihat pada sub bab 2.3. Koefisien-koefisien tersebut berisi aproksimasi dan detail dari masing-masing level hasil dekomposisi level sebelumnya. Hasil dari setiap aproksimasi dan detail ini akan diekstraksi fiturnya di setiap koefisien hasil dekomposisi sinyal tersebut. Proses dekomposisi sinyal dan ekstraksi fitur dapat dilihat pada Kode Sumber 4.3

```

1  for i = 1:length(datatraining)
2      fileDat = load(tmp);
3
4      %dekomposisi sinyal menggunakan wpd
5      T = wpdec(fileDat,3,'db4');
6
7      %mengambil hasil dekomposisi tiap node
8      cfs1 = wpcoef(T,[1 0]);
9      cfs2 = wpcoef(T,[1 1]);
10     cfs3 = wpcoef(T,[2 0]);
11     cfs4 = wpcoef(T,[2 1]);
12     cfs5 = wpcoef(T,[2 2]);
13     cfs6 = wpcoef(T,[2 3]);
14     cfs7 = wpcoef(T,[3 0]);
15     cfs8 = wpcoef(T,[3 1]);
16     cfs9 = wpcoef(T,[3 2]);
17     cfs10 = wpcoef(T,[3 3]);
18     cfs11 = wpcoef(T,[3 4]);
19     cfs12 = wpcoef(T,[3 5]);
20     cfs13 = wpcoef(T,[3 6]);
21     cfs14 = wpcoef(T,[3 7]);
22
23     %ekstraksi fitur
24     wpdtraining(i,1) = mean(cfs1);
25     wpdtraining(i,2) = max(cfs1);
26     wpdtraining(i,3) = min(cfs1);

```

Kode Sumber 4.3 Dekomposisi sinyal WPD

27	<code>wpdtraining(i,4) = std(cfs1);</code>
28	<code>.....</code>
29	<code>.....</code>
30	<code>.....</code>
31	<code>wpdtraining(i,53) = mean(cfs14);</code>
32	<code>wpdtraining(i,54) = max(cfs14);</code>
33	<code>wpdtraining(i,55) = min(cfs14);</code>
34	<code>wpdtraining(i,56) = std(cfs14);</code>
35	<code>end</code>

Kode Sumber 4.4 Dekomposisi sinyal WPD Lanjutan

Pada Kode Sumber 4.3, sinyal didekomposisi menggunakan fungsi *Daubechies* 4 dengan level dekomposisi sebanyak 3 level seperti yang ditunjukkan pada baris ke 5. Pada baris ke 8 sampai ke 21 digunakan untuk mendapatkan nilai aproksimasi dan detail dari masing-masing node. Selanjutnya nilai dari masing-masing aproksimasi dan detail tersebut diekstraksi fiturnya dengan mencari nilai rata-rata, maximum, minimum, dan standar deviasi seperti yang ditunjukkan pada baris ke 23 sampai 35. Data hasil keluaran WPD pada kode sumber diatas adalah matriks baru hasil ekstraksi fitur. Pada Tugas Akhir ini digunakan dekomposisi sinyal sebanyak 3 level sehingga menghasilkan 56 fitur baru.

4.2.3 Implementasi Principal Component Analysis (PCA)

Proses selanjutnya setelah mendekomposisi sinyal dan ekstraksi fitur adalah mereduksi data fitur menggunakan *Principal Component Analysis* (PCA). Pada Tugas Akhir ini, implementasi PCA menggunakan *library* dari perangkat lunak MATLAB R2014a yaitu fungsi *princomp()*. Parameter input dari fungsi *princomp()* merupakan data hasil ekstraksi fitur WPD. Sedangkan parameter output yang dihasilkan berupa nilai loading (COEFF), nilai hasil skor komponen (SCORE) dan nilai eigen (LATENT). Penjelasan lebih lanjut tentang PCA dapat dilihat pada sub Bab 3.2.


```

1  %proses PCA data training
2  [COEFF,SCORE,LATENT] = pca(traindata);
3
4  %proses PCA data testing
5  SCOREtest = testdata * COEFF;
6
7  [rowpcatrain,colpcatrain] = size(SCORE);
8
9  %menyimpan hasil PCA training LATENTtrain>1
10 for i=1:rowpcatrain
11     for j=1:28
12         datapcatraining(i,j) = SCOREtrain(i,j);
13     end
14 end

```

Kode Sumber 4.5 Reduksi dimensi PCA

Hasil dari Kode Sumber 4.4 merupakan data dengan dimensi yang rendah baik untuk data training maupun data testing. Untuk data testing, proses reduksi dimensi menggunakan nilai loading (COEFF) dari data training seperti yang ditunjukkan pada baris ke 5. Setelah reduksi dimensi PCA diimplementasikan, selanjutnya data dianggap sudah siap untuk masuk ke proses klasifikasi.

4.2.4 Implementasi Kombinasi Genetic Algorithm dan Neural Network

Setelah seluruh praproses dalam sistem Tugas Akhir ini telah dilakukan, selanjutnya sistem akan masuk ke tahap klasifikasi menggunakan *Neural Network* yang dioptimasi menggunakan *Genetic Algorithm* dalam pembaharuan bobot dan bias awal. Oleh karena itu, setelah algoritma *feedforward* telah dijalankan, sistem akan melakukan pembaharuan pada bobot dan bias *network* yang dibantu menggunakan metode *Genetic Algorithm* untuk mencari bobot dan bias optimal.

Seperti yang dijelaskan pada sub bab 3.4.1, *Neural Network* yang digunakan pada Tugas Akhir ini terdapat 3 layer, yaitu satu input layer, satu hidden layer, dan satu output layer.

Terdapat 2 node pada output layer, yaitu alkoholik atau control. Sedangkan jumlah node pada hidden layer dibatasi hanya terdapat 15 node dengan mempertimbangkan kompleksitas sistem. Proses membuat model *network* pada proses klasifikasi menggunakan *Neural Network* dapat dilihat pada Kode Sumber 4.5.

```

1  %pembangunan network
2  net = newff(training,target, [15],
3  {'logsig','logsig'});
4  net.trainParam.lr = 0.01;
5  net.trainParam.mc = 0.35;
6  net.trainParam.epochs = 100;
7
8  %training data
9  [net] = configure(net,training,target);
10 [net] = train(net,training,target);
11
12 %mendapatkan bobot dan bias
13 weightBias = getwb(net);
14 [y,z] = size(weightBias);
15
16 %testing data untuk mendapatkan MSE
17 testY = net(testing);
18 mses = mse(net,targettest,testY);
19 msets = 1/mses;

```

Kode Sumber 4.6 Pembangunan Neural Network

Pada Kode Sumber 4.5, baris ke 1 sampai ke 6 adalah perancang model *network* baru untuk proses training data yang ditunjukkan pada kode sumber 8-10. Selanjutnya baris ke 12-14 digunakan untuk mendapatkan bobot dari *network* yang akan dioptimasi menggunakan metode GA pada proses selanjutnya. Sedangkan baris ke 16-18 merupakan proses testing data untuk mendapatkan MSE sebagai nilai evaluasi dari *network* tersebut. Proses diatas kemudian dilakukan sebanyak jumlah populasi yang diinginkan untuk ke proses selanjutnya yaitu GA.

Setelah mendapatkan bobot dan bias hasil *network* diatas, selanjutnya sistem akan masuk ke proses dari GA yaitu *selection*. Seperti yang telah dijelaskan sebelumnya, pada Tugas Akhir ini

proses *selection* dari GA menggunakan *Roulette Wheel Selection* (RWS). Implementasi *selection* dapat dilihat pada Kode Sumber 4.6

```

1  %roulette wheel
2  tempProb=zeros(1,maxpop);
3  temp=0;
4  for k= 1:maxpop
5      tempProb(k) = temp + fitnessfunc(k);
6      temp = tempProb(k);
7  end
8
9  %proses random pemilihan parent
10 random1 = double(rand*(tempProb(end)));
11 index1 = find(tempProb >= random1, 1 );
12
13 index2 = index1;
14
15 while index1 == index2
16     random2 = rand*(tempProb(end));
17     index2 = find(tempProb >= random2, 1 );
18 end

```

Kode Sumber 4.7 Selection

Kode Sumber 4.6 merupakan proses pemilihan *parent* awal menggunakan metode *Roulette Wheel* untuk proses GA. Pada baris ke 1 sampai ke 7 adalah parameter awal yang ditujukan untuk pembuatan kemungkinan parent tersebut dipilih berdasarkan *fitness function* dari pembuatan *network* pada proses sebelumnya. Sedangkan baris ke 9 – 11 merupakan proses pemilihan parent index secara random. Baris ke 15-18 digunakan apabila parent index pertama dan parent index kedua berasal dari index yang sama, sehingga parent kedua akan kembali diacak secara random hingga mendapatkan parent index yang berbeda dari parent index pertama. *Selection* sendiri merupakan proses awal dari proses penting GA setelah inisialisasi. Setelah implementasi *selection*, proses selanjutnya adalah implementasi *crossover* yang dapat dilihat pada Kode Sumber 4.7.

```

1  %crossover
2  CMin = z;
3  CMax = y;
4  alpha = 0.9;
5
6  crossoverpoint = floor(CMin + (CMax-
7  CMin).*rand(1,1));
8
9  tempweight1 = kromosom(:,index1);
10 tempweight2 = kromosom(:,index2);
11 tempweight11 = tempweight1;
12 tempweight22 = tempweight2;
13
14 for j=crossoverpoint:CMAX
15     tempweight1(j) = alpha*tempweight22(j) + (1-
16 alpha)*tempweight11(j);
17     tempweight2(j) = alpha*tempweight11(j) + (1-
18 alpha)*tempweight22(j);
19 end

```

Kode Sumber 4.8 Crossover

Pada Kode Sumber 4.7 adalah mengambil kromosom dari index yang terpilih pada proses *selection*. Kromosom pada GA disini berupa bobot dan bias awal dari *network* tersebut. Baris ke 6-7 adalah proses penentuan *crossover point* sedangkan baris ke 14-19 adalah proses dari *crossover* itu sendiri. Hasil dari proses *crossover* adalah kromosom yang sudah dengan nilai yang berbeda dengan parent sebelumnya.

```

1  %mutasi
2  mutationprob = 0.095;
3
4  for j=1:CMAX
5      if rand < mutationprob
6          tempweight1(j) = rand;
7      end
8  end

```

Kode Sumber 4.9 Mutasi

Setelah proses *crossover*, proses selanjutnya dari GA yaitu mutasi. Kode Sumber 4.8 adalah proses mutasi dari GA. Pada baris ke 2 merupakan parameter dari probabilitas mutasi tersebut. Sedangkan baris ke 4-8 merupakan proses utama dari proses mutasi untuk kromosom-kromosom hasil dari proses sebelumnya. Hasil dari Kode Sumber 4.8 adalah kromosom-kromosom setelah proses GA dilakukan. Selanjutnya yaitu proses evaluasi dimana kromosom-kromosom yang merupakan bobot dan bias awal dari *network* tersebut dievaluasi. Proses evaluasi bobot awal dari GA ini sebenarnya sama saja dengan proses traning *Neural Network* sebelumnya. Hanya saja, pada proses evaluasi disini, bobot dan bias yang digunakan merupakan hasil dari proses-proses GA yang sudah diimplementasikan sebelumnya. Proses evaluasi pada GA dapat dilihat pada Kode Sumber 4.9.

```

1  %inisialisasi network
2  net = newff(training, target, [15],
3  {'logsig','logsig'});
4  net.trainParam.lr = 0.01;
5  net.trainParam.mc = 0.35;
6  net.trainParam.epochs = 100;
7
8  %set weight dan bias setelah GA
9  [net] = configure(net,training,target);
10 net = setwb(net,tempweighth1);
11 [net] = train(net,training,target);
12
13 %testing data untuk mendapatkan MSE
14 testY = net(testing);
15 mses = mse(net,targettest,testY);
16 mseGA = 1/mses;

```

Kode Sumber 4.10 Evaluasi GA

Baris ke 1 sampai ke 6 merupakan inisialisasi *network*. baris ke 8-11 merupakan proses untuk menginisialisasi bobot dan bias awal setelah GA yang sudah diimplementasikan pada proses sebelumnya. Baris ke 13 -16 merupakan proses testing untuk mendapatkan nilai MSE dari *network* tersebut yang akan

dibandingkan dengan nilai MSE dari parent sebelumnya pada proses selanjutnya yaitu evaluasi data testing. Proses evaluasi data testing dapat dilihat pada Kode Sumber 4.10.

```

1      %elitis
2      msekromosomga= mseGA;
3      if msekromosomga>fitnessfunc(index1)
4          kromosom(:,index1)=tempweigth1;
5          fitnessfunc(index1)=mseGA;
6      end
7
8      %mencari kromosom terbaik untuk testing
9      for m=1:maxpop
10         if fitnessfunc(m)>min
11             min=fitnessfunc(m);
12             ind=m;
13         end
14     end
15
16     %mengganti kromosom terbaik
17     bestweightBias = kromosom(:,ind);
18
19     %inisialisasi network
20     net = newff(training, target, [15],
21         {'logsig','logsig'});
22     net.trainParam.lr = 0.01;
23     net.trainParam.mc = 0.35;
24     net.trainParam.goal = 0;
25
26     %inisialisasi weight setelah GA
27     [net] = configure(net,training,target);
28     net = setwb(net,bestweightBias);
29
30     %testing network
31     testY = net(testing);
32
33     %mencari accuracy, sensitivity, spesititivity
34     [~,cm] = confusion(targettest,testY);
35     Accuracy =
36         (cm(1,1)+cm(2,2))/(cm(1,1)+cm(2,1)+cm(1,2)+cm(2,2))*100;
37
38     Sensitivity = (cm(1,1)/(cm(1,1)+cm(1,2)))*100;
39     Spesititivity = (cm(2,2)/(cm(2,2)+cm(2,1)))*100;

```

Kode Sumber 4.11 Evaluasi Data Testing

Pada Kode Sumber 4.10 ditunjukkan hal pertama yang dilakukan adalah membandingkan nilai MSE dari parent dan hasil GA. Apabila kromosom baru hasil GA tersebut menghasilkan MSE yang lebih baik dari parentnya, maka kromosom pada index parent sebelumnya akan digantikan dengan kromosom hasil GA. Proses ini juga dapat disebut dengan proses *elitis*. Hal tersebut ditunjukkan untuk mempertahankan populasi terbaik yang nantinya akan dihasilkan oleh GA. Setelah mendapatkan populasi terbaik dari proses GA, kromosom dengan nilai MSE terkecil yang akan digunakan untuk implementasi proses testing. Cara kerja evaluasi data testing sendiri, GA nantinya akan mencari index dari kromosom dengan nilai MSE yang paling kecil. Hal ini dilakukan karena *network* dengan MSE paling kecil merupakan *network* terbaik yang dihasilkan. Selanjutnya adalah mencari bobot dan bias terbaik hasil *elitist* dengan MSE terkecil seperti yang ditunjukkan pada baris ke 8 sampai 17. Selanjutnya baris ke 19 sampai 24 merupakan proses inisialisasi *network* dan baris ke 26-28 merupakan inisialisasi bobot dan bias hasil GA terbaik pada *network*. Baris ke 33-39 adalah proses testing. Sedangkan baris ke 26-32 digunakan untuk menghitung performa berupa akurasi, sensitivitas, dan spesifikasi dari sistem.

BAB V

UJI COBA DAN EVALUASI

Bab ini akan dijelaskan mengenai skenario uji coba pada perangkat lunak yang telah dibangun. Setelah itu, hasil uji coba akan dievaluasi kinerjanya sehingga dapat diputuskan apakah perangkat lunak ini mampu menyelesaikan permasalahan yang telah dirumuskan diawal. Secara garis besar, bab ini berisikan pembahasan mengenai lingkungan pengujian, data pengujian, dan uji kinerja.

5.1 Lingkungan Pengujian

Lingkungan pengujian pada uji coba permasalahan Klasifikasi EEG Alkoholik Menggunakan *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan Kombinasi *Genetic Algorithm* dan *Neural Network*. menggunakan spesifikasi keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Uji Coba

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz 2.39 GHz
	Memori	8 GB 1600
Perangkat Lunak	Sistem Operasi	Windows 10 Pro
	Perangkat Pengembang	MATLAB R2014a

5.2 Data Pengujian

Seperti yang telah dijelaskan sebelumnya, data yang digunakan pada Tugas Akhir ini adalah data sinyal otak manusia (EEG) yang berasal dari *Neurodynamics Laboratory, State University of New York Health Centre*. Data ini diunduh dari website *UCI Machine Learning Repository*. Pada Tugas Akhir ini, data awalan adalah data acak yang diambil dari sampel yang berbeda. Sedangkan untuk skenario uji coba, dataset akan dibagi menjadi 2 data yang berbeda yaitu:

1. Dataset 1 merupakan data acak yang terdiri dari 60 training dan 40 testing yang diambil sama dengan paper acuan [15].
2. Dataset 2 merupakan data acak yang terdiri dari 120 training dan 40 testing.

Data awalan ini kemudian akan masuk ke proses dekomposisi dan ekstraksi fitur menggunakan WPD sehingga menghasilkan matrix yang berbeda antara data training dan data testing. Selanjutnya data hasil ekstraksi fitur WPD direduksi dimensinya menggunakan PCA sehingga menghasilkan fitur baru yang lebih sedikit dari hasil ekstraksi fitur WPD. Setelah melewati proses PCA, data sudah dianggap siap untuk masuk ke proses klasifikasi menggunakan GA-NN.

5.3 Skenario Uji Coba

Sebelum melakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario uji coba ini, perangkat akan diuji apakah sudah berjalan dengan benar, bagaimana performa pada masing-masing skenario dan perbandingan performa antara skenario mana yang memiliki hasil paling baik. Pada Tugas Akhir ini, terdapat berbagai macam skenario uji coba, yaitu:

1. Perhitungan performa berdasarkan jumlah fitur yang dihasilkan *Principal Component Analysis*.
2. Perbandingan performa dengan atau tanpa *Genetic Algorithm* (GA) pada metode *Neural Network*
3. Perhitungan performa berdasarkan parameter *Learning Rate* pada klasifikasi *Neural Network*.
4. Perhitungan performa berdasarkan parameter dari GA yaitu nilai alpha pada metode *crossover*.
5. Perbandingan performa antara metode yang diusulkan dari penelitian sebelumnya dan metode yang diusulkan pada Tugas Akhir ini.

5.3.1 Skenario Uji Coba Perhitungan Performa Berdasarkan Fitur yang dihasilkan PCA

Skenario uji coba pertama adalah perhitungan performa berdasarkan jumlah fitur yang dihasilkan *Principal Component Analysis* (PCA). Pada dasarnya, hasil dari PCA sendiri adalah fitur baru dengan dimensi yang lebih kecil. Hal ini kemudian akan menyebabkan jumlah fitur awal dengan fitur hasil PCA menjadi berbeda. Oleh karena itu, pada skenario uji coba ini performa PCA akan dihitung berdasarkan persentase jumlah fitur yang diambil. Pengambilan fitur setelah reduksi dimensi PCA sendiri dengan memprioritaskan fitur dengan nilai *eigen value* tertinggi. Banyaknya jumlah fitur yang diambil setelah metode PCA dapat dilihat pada Tabel 5.2.

Tabel 5.2 Jumlah Fitur yang Dihasilkan PCA

Banyak Fitur	Jumlah Fitur pada Dataset 1 & 2
30%	17
40%	23
50%	28
60%	34

Perhitungan performa dari PCA sendiri dengan menghitung akurasi yang dihasilkan dari masing-masing dataset 1 maupun dataset 2. Pada skenario uji coba 1, data sebelumnya didekomposisi sebanyak 3 level menggunakan WPD dan selanjutnya akan diklasifikasi menggunakan Kombinasi *Genetic Algorithm* dan *Neural Network* dengan nilai *Learning Rate* 0.1 dan nilai alpha pada metode *crossover* 0.9. Hasil performa uji coba dari PCA dapat dilihat pada Tabel 5.3.

Tabel 5.3 Perhitungan Performa PCA

Banyak Fitur	Rata-rata Akurasi (%)	
	Dataset 1	Dataset 2
30%	89.00	81.50
40%	84.50	78.00
50%	80.50	77.00
60%	79.00	73.50

Kesimpulan dari hasil uji coba yang ditunjukkan pada tabel perhitungan performa dari PCA diatas adalah semakin sedikit fitur yang diambil, semakin besar akurasi yang dihasilkan. Hal ini disebabkan karena bobot dan bias yang diinisialisasi random tersebut menghasilkan akurasi yang berbeda setiap percobaan meskipun perbedaannya tidak terlalu jauh. Hasil dari setiap percobaan dan perhitungan rata-rata dari Tabel 5.3 dapat dilihat pada sub bab lampiran.

5.3.2 Skenario Uji Coba Performa Metode Genetic Algorithm Pada Neural Network

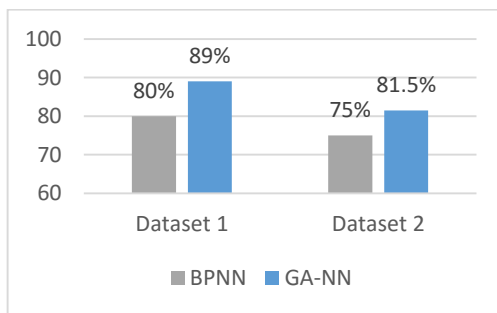
Pada skenario uji coba yang kedua akan diuji performa dari metode *Genetic Algorithm* (GA) pada metode *Neural Network* dalam mengoptimasi bobot dan bias awal sistem klasifikasi. Perhitungan performa sendiri dengan menghitung akurasi dan *running-time* dari *Neural Network* tanpa GA dan GA-NN. Selain itu, proses perhitungan performa juga diuji berdasarkan

konvergensi yang dihasilkan dari kedua metode tersebut. Pada skenario uji coba ini data yang digunakan adalah dataset setelah reduksi dimensi PCA dengan banyak fitur sebanyak 30%. Hasil performa dari kombinasi metode *Genetic Algorithm* dan *Neural Network* dapat dilihat pada Tabel 5.4.

Tabel 5.4 Perbandingan Performa BPNN dan GA-NN

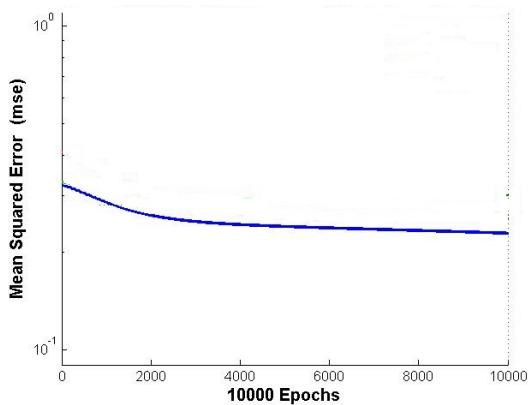
Klasifikasi	Akurasi (%)		Running-time (s)	
	Dataset 1	Dataset 2	Dataset 1	Dataset 2
BPNN	80.00	75.00	44.000	45.000
GA-NN	89.00	81.50	45.008	45.451
Selisih	9.00	6.50	1.008	0.451

Berdasarkan hasil uji coba yang ditunjukkan pada Tabel 5.4 dapat ditarik kesimpulan bahwa metode kombinasi GA-NN dapat menaikkan tingkat akurasi sebesar 9.00% untuk dataset 1 dan 6.50% untuk dataset 2 dibandingkan dengan metode *Neural Network* tanpa GA. Akan tetapi, hal sebaliknya terjadi pada hasil *Running-time* yang dihasilkan menunjukkan metode kombinasi GA-NN memiliki hasil *running-time* yang tinggi dibandingkan *Neural Network* tanpa GA. Hal ini tentunya disebabkan karena proses-proses dari GA sendiri memiliki kompleksitas yang cukup tinggi sehingga membutuhkan waktu komputasi yang lebih lama.

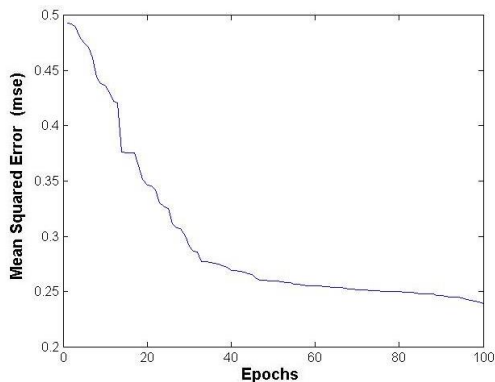


Gambar 5.1 Perbandingan Akurasi BPNN dan GA-NN

Selain, menghitung akurasi yang dihasilkan metode BPNN dan kombinasi GA-NN, perhitungan performa berdasarkan konvergensi yang dihasilkan dari kedua metode juga dapat dilakukan. Proses perhitungan performa berdasarkan konvergensi yang dihasilkan juga dengan menentukan jumlah iterasi pada BPNN untuk mencapai konvergensi. Oleh karena itu, jumlah iterasi pada BPNN ditentukan sebanyak 10000 iterasi. Perhitungan performa berdasarkan pencapaian konvergensi dari kedua metode dapat dilihat pada Gambar 5.2 dan 5.3.



Gambar 5.2 Grafik MSE BPNN



Gambar 5.3 Grafik MSE Kombinasi GA-NN

Seperti yang ditunjukkan pada Gambar 5.2 dan 5.3, metode BPNN dan kombinasi GA-NN belum mencapai proses konvergensi hingga mencapai kondisi pemberhentian. Hal ini disebabkan karena kedua metode tersebut masih membutuhkan jumlah iterasi yang lebih banyak untuk dapat mencapai konvergensi. Akan tetapi dengan mempertimbangkan kompleksitas sistem dan waktu komputasi, kondisi pemberhentian pun ditentukan diawal dengan penentuan pada metode BPNN sebanyak 10.000 iterasi dan metode kombinasi GA-NN sebanyak 100 iterasi. Selain itu, dari skenario uji coba ini dapat ditarik kesimpulan bahwa kombinasi GA-NN dengan 100 iterasi dapat menghasilkan MSE yang lebih kecil dibandingkan dengan metode BPNN yang mencapai 10.000 iterasi.

5.3.3 Skenario Uji Coba Perhitungan Performa Berdasarkan Learning Rate NN

Pada uji coba skenario yang ketiga akan dilakukan perhitungan performa pada metode klasifikasi *neural network* dengan mengganti konstanta dari parameter *learning rate* yang diinisialisasi di awal. Parameter *learning rate* merupakan parameter yang digunakan sebagai nilai pembelajaran *neural network* untuk mengontrol pembaharuan bobot dan bias hingga mencapai konvergen. Parameter yang akan digunakan adalah 0.01, 0.05, 0.1 dan 0.2. Perhitungan performa pada skenario uji coba ini dengan menghitung rata-rata akurasi yang dihasilkan dari 5 percobaan berdasarkan nilai *learning rate* yang ditentukan. Dataset yang digunakan adalah data setelah PCA dengan jumlah fitur 30.00%. Sedangkan metode klasifikasi menggunakan metode kombinasi GA-NN dengan nilai alpha pada proses GA 0.8. Hasil perhitungan performa setiap nilai awal parameter *Learning Rate* dapat dilihat pada Tabel 5.5

Tabel 5.5 Perhitungan Performa GA-NN Berdasarkan Learning Rate

Nilai <i>Learning Rate</i>	Rata-rata Akurasi (%)	
	Dataset 1	Dataset 2
0.01	86.50	80.00
0.05	88.00	82.50
0.1	89.00	81.50
0.2	86.50	79.00

Berdasarkan hasil uji coba yang ditunjukkan pada Tabel 5.5 dapat ditarik kesimpulan bahwa rata-rata akurasi tertinggi yang dihasilkan berdasarkan nilai *learning rate* 0.1 pada dataset 1 sebesar 89.00% dan nilai 0.05 pada dataset 2 sebesar 82.50%. Hasil akurasi dari setiap percobaan dan perhitungan rata-rata akurasi yang dihasilkan dari Tabel 5.5 dapat dilihat pada sub bab lampiran.

5.3.4 Skenario Uji Coba Perhitungan Performa kombinasi GA-NN Berdasarkan Nilai α

Pada skenario uji coba ke empat, akan dilakukan perhitungan performa pada metode optimasi GA. Parameter uji coba yang digunakan adalah nilai α (α), yaitu konstanta pada *arithmetic crossover*. Pengaruh nilai α (α) pada *arithmetic crossover* adalah nilai pada kromosom yang terdapat pada *child* hasil *crossover* tersebut. Dengan kata lain, semakin kecil nilai konstanta α (α), nilai kromosom tersebut akan semakin sama dengan nilai kromosom pada *parent*. Sebaliknya, semakin besar nilai konstanta α (α), semakin berbeda nilai kromosom pada *child* tersebut dengan nilai kromosom pada *parent*. Perhitungan performa pada sendiri dengan menghitung nilai rata-rata akurasi yang dihasilkan dari masing-masing nilai konstanta tersebut. Pada skenario uji coba inidigunakan fitur setelah PCA 30.00% dan parameter *learning rate* NN yang terbaik untuk masing-masing dataset. Pengujian pada konstanta α dapat dilihat pada Tabel 5.6.

Tabel 5.6 Perhitungan performa GA-NN Berdasarkan Nilai α

Nilai α	Rata-rata Akurasi (%)	
	Dataset 1	Dataset 2
0.6	82.50	84.50
0.7	87.00	81.00
0.8	89.00	82.50
0.9	94.00	82.50

Berdasarkan hasil uji coba yang ditunjukkan pada Tabel 5.6 dari perhitungan performa kombinasi GA-NN berdasarkan nilai α dapat ditarik kesimpulan bahwa rata-rata akurasi tertinggi yang dihasilkan dataset 1 sebesar 94.00% dengan nilai α 0.9 dan 82.50% untuk dataset 2 dengan nilai α 0.6. Hasil akurasi dari setiap percobaan dan perhitungan rata-rata akurasi yang dihasilkan dari Tabel 5.5 dapat dilihat pada sub bab lampiran.

5.3.5 Skenario Uji Coba Perbandingan Performa dengan Penelitian Sebelumnya

Pada skenario uji coba ke lima, akan dilakukan perbandingan antara metode yang diusulkan pada Tugas Akhir ini dengan metode pada penelitian sebelumnya. Penelitian tersebut adalah klasifikasi alkoholik dari data EEG menggunakan dataset yang sama dengan dataset 1 pada Tugas Akhir ini, akan tetapi menggunakan metode yang berbeda yaitu *Independent Component Analysis* (ICA), *Wavelet de-noising*, dan *Probabilistic Neural Network* (PNN) [15]. Oleh karena itu, pada skenario uji coba ini hanya membandingkan akurasi terbaik yang dihasilkan dari dataset 1 saja dari ke empat skenario uji coba sebelumnya. Adapun perbandingan hasil akurasi terbaik yang didapatkan dari penelitian sebelumnya dan Tugas Akhir ini dapat dilihat pada Tabel 5.7.

Tabel 5.7 Perhitungan Performa dengan Penelitian Sebelumnya

Metode yang digagaskan	Akurasi Terbaik (%)
Penelitian sebelumnya [15]	85.00
Metode yang diusulkan	94.00
Selisih	9.00

Berdasarkan hasil uji coba yang ditunjukkan pada Tabel 5.7, dapat ditarik kesimpulan bahwa hasil akurasi terbaik yang dihasilkan menggunakan metode yang diusulkan pada Tugas Akhir ini yaitu *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan Kombinasi *Genetic Algorithm* dan *Neural Network* dapat menghasilkan akurasi yang lebih baik hingga mencapai selisih 9.00% menggunakan dataset yang sama dibandingkan dengan metode pada penelitian sebelumnya yaitu *Independent Component Analysis* (ICA), *Wavelet de-noising*, dan *Probabilistic Neural Network* (PNN).

5.4 Evaluasi Umum Skenario Uji Coba

Berdasarkan skenario uji coba yang telah dilakukan dengan mengganti nilai parameter awal, dapat diketahui bahwa sistem Klasifikasi EEG Alkoholik menggunakan *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan Kombinasi *Genetic Algorithm* dan *Neural Network* mampu mendeteksi alkoholik dengan baik dengan rata-rata akurasi paling besar 94.00% oleh dataset 1 yang terdiri dari 60 data training dan 40 data testing menggunakan fitur PCA 30%, parameter *learning rate* 0.05, dan nilai *alpha* (α) pada proses *crossover* 0.9. Sedangkan pada dataset 2 yang terdiri dari 120 data training dan 40 testing menghasilkan rata-rata akurasi tertinggi 84.50% menggunakan fitur PCA 30.00%, parameter *learning rate* 0.1 dan nilai *alpha* (α) pada *crossover* 0.6. Selain itu, Metode metode yang diusulkan pada Tugas Akhir ini dapat menaikkan akurasi hingga 9.00% dibandingkan dengan metode *Independent Component Analysis*

(ICA), *Wavelet de-noising*, dan *Probabilistic Neural Network* (PNN) yang digagaskan pada penelitian sebelumnya.

Pada skenario uji coba 1 didapatkan bahwa rata-rata akurasi terbaik yang dihasilkan dengan mengambil fitur hasil PCA sebesar 30% baik untuk dataset 1 maupun dataset 2. Pada skenario uji coba 2 dapat diketahui bahwa metode GA sendiri dapat membantu algoritma NN dalam mengoptimasi bobot dan bias yang diset awal dengan rata-rata kenaikan akurasi sebesar 9.00% dan 6.50% untuk masing-masing dataset 1 dan 2. Akan tetapi, metode NN menggunakan algoritma GA membutuhkan waktu komputasi lebih besar 44 detik dibandingkan BPNN biasa. Selanjutnya pada skenario uji coba 3 mengambil hasil fitur hasil PCA sebesar 30% serta mengganti nilai parameter *learning rate* pada proses klasifikasi menggunakan metode kombinasi GA-NN hingga menghasilkan akurasi terbaik pada dataset 1 sebesar 89.00% dan dataset 2 82.50%. Hasil rata-rata akurasi yang didapatkan pada skenario uji coba ke 4 menghasilkan akurasi tertinggi sebesar 94.00% untuk dataset 1 dan juga menghasilkan akurasi sebesar 84.50% untuk dataset 2. Pada skenario uji coba terakhir, digunakan hasil fitur hasil PCA 30% untuk kedua data awalan, nilai *learning rate* 0.05 untuk dataset 1 dan 0.1 untuk dataset 2, serta mengganti nilai *alpha* (α) pada metode *arithmetic crossover*. Sedangkan dari skenario uji coba ke 5, dapat ditarik kesimpulan bahwa pada Tugas Akhir ini dengan menggunakan metode *Wavelet Packet Decomposition*, *Principal Component Analysis*, dan Kombinasi *Genetic Algorithm* dan *Neural Network* dapat menaikkan akurasi hingga mencapai 9.00% dibandingkan dengan metode yang digagaskan pada penelitian sebelumnya yaitu *Independent Component Analysis* (ICA), *Wavelet de-noising*, dan *Probabilistic Neural Network* (PNN).

(Halaman ini sengaja dikosongkan)

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

6.1 Kesimpulan

Kesimpulan yang dapat diambil didapatkan berdasarkan hasil uji coba Klasifikasi Data EEG Alkoholik Menggunakan *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan Kombinasi Metode *Genetic Algorithm* dan *Neural Network* adalah sebagai berikut:

1. Metode *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan klasifikasi menggunakan Kombinasi *Genetic Algorithm* dan *Neural Network* dapat digunakan untuk ekstraksi fitur dan klasifikasi data EEG alkoholik dengan hasil rata-rata akurasi tertinggi sebesar 94.00%.
2. Berdasarkan hasil uji coba, metode *Principal Component Analysis* menghasilkan rata-rata akurasi tertinggi sebesar 89.00% untuk dataset 1 dan 81.50% untuk dataset 2 dengan pengambilan fitur sebesar 30%.
3. Implementasi *Genetic Algorithm* (GA) dapat dijadikan algoritma optimasi untuk metode *Back Propagation Neural Network* yang cocok untuk mengoptimasi bobot dan bias awal dengan menaikkan akurasi 9.00% untuk dataset 1 dan 6.50% untuk dataset 2. Akan tetapi membutuhkan waktu komputasi yang lebih lama hingga mencapai 45 detik.

4. Rata-rata akurasi tertinggi yang didapat dari skenario uji coba adalah 94.00% untuk dataset 1 dengan parameter *learning rate* 0.1 dan nilai *alpha* 0.9. Sedangkan untuk dataset 2 menghasilkan rata-rata akurasi paling tinggi 84.50% dengan parameter *learning rate* 0.05 dan nilai *alpha* 0.6.
5. Klasifikasi EEG Alkoholik menggunakan *Wavelet Packet Decomposition* (WPD), *Principal Component Analysis* (PCA), dan Kombinasi *Genetic Algorithm* dan *Neural Network* dapat menghasilkan akurasi yang lebih baik dari metode *Independent Component Analysis* (ICA), *Wavelet de-noising*, dan *Probabilistic Neural Network* (PNN) hingga mencapai 9.00%.

6.2 Saran

Saran yang diberikan terkait pengembangan pada Tugas Akhir ini adalah:

1. Sistem sebaiknya dilengkapi dengan GUI untuk memudahkan penggunaan.
2. Menggunakan level dekomposisi sinyal WPD lebih dari 3 level.
3. Dari segi klasifikasi, kombinasi metode *Genetic Algorithm* dan *Neural Network* membutuhkan banyak parameter yang diinisialisasi di awal sehingga disarankan untuk menggunakan metode klasifikasi lain.

DAFTAR PUSTAKA

- [1] M. R. N. Kousarrizi, A. G. A., G. A, M. Teshnehlal dan A. M., "Classification of Alcoholics and Non-Alcoholics via EEG Using SVM and Neural Networks," *IEEE*, 2009.
- [2] Shuaifang Wang, Y. Li, P. Wen, dan D. Lai, "Data selection in EEG signals classification," dalam *Australasian Physical & Engineering Sciences in Medicine*, 2016, pp. 157-165.
- [3] M. M. Shaker, "EEG Waves Classifier using Wavelet Transform and Fourier Transform," *International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering*, vol. 1, 2007.
- [4] A. Turnip, K.-S. Hong dan S. S. Ge, "Backpropagation neural networks training for single trial EEG classification," *IEEE*, 2010.
- [5] J. Kowalewski dan F. Losch, "Teens and Alcohol," 19 April 2010. [Online]. Available: <http://mrkhealth.pbworks.com/w/page/23222837/In%20what%20ways%20can%20teens%20avoid%20the%20effects%20of%20alcohol>. [Diakses 19 Februari 2017].
- [6] "P300 Detection Based on EEG Shape Features," Research Gate, [Online]. Available: https://www.researchgate.net/figure/289984682_fig2_The-electrode-configuration-used-in-the-current-study-The-EEG-was-acquired-by-10. [Diakses 15 05 2017].
- [7] "EEG," [Online]. Available: <http://somatosphere.net/2014/01/eeg.html>. [Diakses 2017 May 15].
- [8] Wikipedia, "Wavelet packet decomposition," 19 July 2016. [Online]. Available: https://en.wikipedia.org/wiki/Wavelet_packet_decomposition. [Diakses 19 February 2017].

- [9] “Wavelet Packets,” MathWorks, [Online]. Available: <https://www.mathworks.com/help/wavelet/ug/wavelet-packets.html>. [Diakses 19 February 2017].
- [10] “A Beginner's Guide to Eigenvectors, PCA, Covariance and Entropy,” Deeplearning4j, [Online]. Available: <https://deeplearning4j.org/eigenvector>. [Diakses 19 February 2017].
- [11] M. A., Gao dan R.X, “PCA-Based Feature Selection Scheme for Machine Defect Classification,” *IEEE*, vol. 53, 2004.
- [12] “Mean squared error,” Wikipedia, 24 February 2017. [Online]. Available: https://en.wikipedia.org/wiki/Mean_squared_error. [Diakses 2017 February 2017].
- [13] S. Wang, S. Yin dan M. Jiang, “Hybird Neural Network Based On GA-BP for Personal Credit Scoring,” *IEEE Computer Society*, 2008.
- [14] K. F. Man, “Genetic Algorithms: Concepts and Applications,” *IEEE*, vol. 43, 1996.
- [15] N. T. Rachman, H. Tjandrasa dan C. Fatichah, “Alcoholism Classification Based on EEG data using Independent Component Analysis (ICA), Wavelet denoising, and Probabilistic Neural Network (PNN),” *IEEE*, no. International Seminar on Intelligent Technology and Its Aoplications, 2016.
- [16] R. Rojas, “Neural Networks,” Berlin, Springer-Verlag, 1996, pp. 151-184.
- [17] A. Amin, “A novel classification model for cotton yarn quality based on trained neural network using genetic algorithm,” *Science Direct*, 2012.
- [18] M. Y. Gokhale dan D. K. Khanduja, “Time Domain Signal Analysis Using Wavelet Packet Decomposition Approach,” *Scientific Research*, p. 321–329, 2010.

- [19] W. Ting a, Y. Guo-zheng, Y. Bang-hua dan S. Hong, “EEG feature extraction based on wavelet packet decomposition for brain computer interface,” *Science Direct*, 2007.
- [20] U. R. Acharya,, S. V. Sree dan A. P. Chu, “Use of principal component analysis for automatic classification of epileptic EEG activities in wavelet framework,” *Science Direct*, 2012.

(Halaman ini sengaja dikosongkan)

LAMPIRAN

A. Data yang Digunakan Untuk Skenario Uji Coba

1. Dataset 1

Training		Testing	
Alkoholik	Normal	Alkoholik	Normal
A1	C1	A31	C41
A2	C2	A32	C42
A3	C3	A33	C43
A4	C4	A34	C44
A5	C5	A35	C45
A6	C6	A36	C46
A7	C7	A37	C47
A8	C8	A38	C48
A9	C9	A39	C49
A10	C10	A40	C40
A11	C11	A41	C41
A12	C12	A42	C42
A13	C13	A43	C43
A14	C14	A44	C44
A15	C15	A45	C45
A16	C16	A46	C46
A17	C17	A47	C47
A18	C18	A48	C48
A19	C19	A49	C49
A20	C20	A50	C50
A21	C21		
A22	C22		
A23	C23		
A24	C24		

A25	C25		
A26	C26		
A27	C27		
A28	C28		
A29	C29		
A30	C30		

2. Dataset 2

Training		Testing	
Alkoholik	Normal	Alkoholik	Normal
A101	C101	A161	C161
A102	C102	A162	C162
A103	C103	A163	C163
A104	C104	A164	C164
A105	C105	A165	C165
A106	C106	A166	C166
A107	C107	A167	C167
A108	C108	A168	C168
A109	C109	A169	C169
A110	C110	A170	C170
A111	C111	A171	C171
A112	C112	A172	C172
A113	C113	A173	C173
A114	C114	A174	C174
A115	C115	A175	C175
A116	C116	A176	C176
A117	C117	A177	C177
A118	C118	A178	C178
A119	C119	A179	C179
A120	C120	A180	C180
A121	C121		

A122	C122		
A123	C123		
A124	C124		
A125	C125		
A126	C126		
A127	C127		
A128	C128		
A129	C129		
A130	C130		
A131	C131		
A132	C132		
A133	C133		
A134	C134		
A135	C135		
A136	C136		
A137	C137		
A138	C138		
A139	C139		
A140	C140		
A141	C141		
A142	C142		
A143	C143		
A144	C144		
A145	C145		
A146	C146		
A147	C147		
A148	C148		
A149	C149		
A150	C150		
A151	C151		
A152	C152		
A153	C153		

A154	C154		
A155	C155		
A156	C156		
A157	C157		
A158	C158		
A159	C159		
A160	C160		

B. Nilai Eigen Value yang Dihasilkan PCA

1. Dataset 1

No.	Nilai Eigen	No.	Nilai Eigen
1.	15209045.48	29.	23.27373622
2.	2473502.576	30.	17.65982919
3.	372411.8516	31.	14.51990182
4.	147348.7784	32.	9.035842925
5.	75663.19083	33.	4.807981191
6.	68612.98442	34.	2.518595446
7.	32902.09665	35.	1.772986891
8.	29969.65163	36.	0.614873563
9.	18319.6534	37.	0.391403264
10.	12306.77908	38.	0.266839989
11.	6536.590535	39.	0.165147497
12.	5477.314858	40.	0.117426703
13.	1769.84622	41.	0.075282976
14.	1066.936523	42.	0.040812309
15.	784.4030942	43.	0.02622784
16.	684.7526948	44.	0.024455807
17.	583.4385211	45.	0.017683981
18.	336.5342522	46.	0.008663431
19.	250.3037021	47.	0.005720344
20.	189.8337613	48.	0.001924928
21.	172.7747655	49.	0.000620273

22.	150.6646019	50.	0.000412483
23.	107.8169592	51.	1.97E-05
24.	71.55781345	52.	2.00E-06
25.	52.35111859	53.	4.04E-07
26.	45.10416328	54.	5.35E-08
27.	30.9054891	55.	4.10E-09
28.	27.09795065	56.	3.00E-24

2. Dataset 2

No.	Nilai Eigen	No.	Nilai Eigen
1.	24510.69065	29.	0.944990499
2.	12446.462	30.	0.723053946
3.	4313.460477	31.	0.544899866
4.	502.7833279	32.	0.100132062
5.	374.4493252	33.	0.096151314
6.	255.0364489	34.	0.08894319
7.	130.0421811	35.	0.060948755
8.	111.1622364	36.	0.015137506
9.	53.49682858	37.	0.009379312
10.	44.77982363	38.	0.003895347
11.	38.64737394	39.	0.003616213
12.	30.72472753	40.	0.001723095
13.	25.91314447	41.	0.001281295
14.	17.03591023	42.	0.000709144
15.	13.21158092	43.	0.000539904
16.	9.479402355	44.	0.000379357
17.	8.862506368	45.	0.000254308
18.	8.405395235	46.	0.000205359
19.	6.933249522	47.	0.000169417
20.	6.394647802	48.	0.000139541
21.	5.099724938	49.	4.24E-05
22.	4.672508163	50.	8.66E-06
23.	3.464064256	51.	2.81E-06

24.	2.569420432	52.	2.83E-07
25.	1.907880006	53.	2.01E-07
26.	1.851626262	54.	1.58E-08
27.	1.477991216	55.	6.21E-09
28.	1.140274327	56.	8.50E-11

C. Hasil Uji Coba dan Perhitungan Rata-Rata

1. Dataset 1 Perhitungan Performa PCA

Banyak Fitur	Akurasi Percobaan ke- (%)					Rata-rata
	1	2	3	4	5	
30%	90	85	92.5	87.5	90	89.00
40%	80	85	87.5	82.5	87.5	84.50
50%	80	75	80	82.5	85	80.50
60%	82.5	80	77.5	80	75	79.00

2. Dataset 2 Perhitungan Performa PCA

Banyak Fitur	Akurasi Percobaan ke- (%)					Rata-rata
	1	2	3	4	5	
30%	80	85	77.5	80	85	81.50
40%	80	80	75	77.5	77.5	78.00
50%	82.5	80	72.5	75	75	77.00
60%	75	75	75	72.5	70	73.50

3. Dataset 1 Perhitungan Performa GA-NN Berdasarkan Learning Rate

Nilai <i>Learning Rate</i>	Akurasi Percobaan ke- (%)					Rata-rata (%)
	1	2	3	4	5	
0.01	87.5	85	87.5	90	82.5	86.50
0.05	85	87.5	90	90	87.5	88.00
0.1	90	85	92.5	87.5	90	89.00
0.2	87.5	87.5	85	87.5	85	86.50

4. Dataset 2 Perhitungan Performa GA-NN Berdasarkan Learning Rate

Nilai <i>Learning Rate</i>	Akurasi Percobaan ke- (%)					Rata-rata (%)
	1	2	3	4	5	
0.01	82.5	85	80	75	77.5	80.00
0.05	82.5	80	87.5	82.5	80	82.50
0.1	80	85	77.5	80	85	81.50
0.2	77.5	77.5	82.5	77.5	80	79.00

5. Dataset 1 Perhitungan Performa GA-NN Berdasarkan Nilai alpha

Nilai <i>alpha</i>	Akurasi Percobaan ke- (%)					Rata-rata (%)
	1	2	3	4	5	
0.6	80	85	80	82.5	85	82.50
0.7	85	85	90	90	85	87.00
0.8	90	85	92.5	87.5	90	89.00
0.9	95	95	92.5	92.5	95	94.00

6. Dataset 2 Perhitungan Performa GA-NN Berdasarkan Nilai alpha

Nilai <i>alpha</i>	Akurasi Percobaan ke- (%)					Rata-rata (%)
	1	2	3	4	5	
0.6	82.5	87.5	85	85	82.5	84.50
0.7	80	80	80	85	80	81.00
0.8	82.5	82.5	87.5	80	80	82.50
0.9	82.5	80	87.5	82.5	80	82.50

BIODATA PENULIS



Muhammad Saddam, lahir di Jakarta pada tanggal 16 November 1995. Penulis merupakan anak ketiga dari tiga bersaudara pasangan Bapak Dahlan dan Ibu Hasniaty. Penulis menempuh pendidikan formal dimulai dari TK Islam Athirah Makassar (1999-2001), SD Islam Athirah Makassar (2001-2004), SD Kasih Ananda I Jakarta (2004-2007), SMP Negeri 30 Jakarta (2007-2010), SMA Negeri 13 Jakarta (2010-2013) dan S1 Teknik Informatika ITS (2013-2017). Bidang studi yang diambil oleh penulis pada saat berkuliah di Teknik Informatika ITS adalah Komputasi Cerdas dan Visi (KCV). Selama masa kuliah, penulis aktif dalam organisasi kemahasiswaan yaitu Himpunan Mahasiswa Teknik Computer-Informatika (2014-2015). Penulis juga aktif dalam berbagai kegiatan kepanitiaan yaitu SCHEMATICS 2014-2015 dan ITS EXPO 2014-2016. Penulis sempat menjadi asisten dosen selama menjalani perkuliahan di Teknik Informatika yaitu asisten dosen mata kuliah Otomata. Penulis memiliki ketertarikan dalam hal *travelling* dan menyukai sesuatu hal yang baru. Komunikasi dengan penulis dapat melalui email: **muhsad.dahlan@gmail.com**